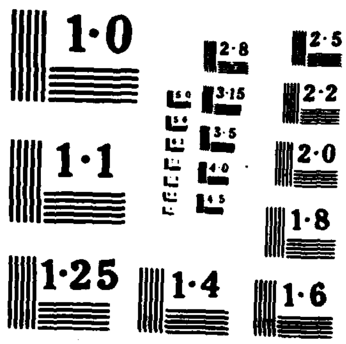
AD-A169 774 CARRIER-SENSE STACK ALGORITHMS FOR MULTIPLE ACCESS COMMUNICATION CHANNELS(U) CONNECTICUT UNIV STORRS DEPT OF ELECTRICAL ENGINEERING AND CO.. L MERAKOS MAR 86 1/1

UNCLASSIFIED UCT/DEECS/TR-86-3 N00014-85-K-0547 F/G 9/2 NL


1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

The University of Connecticut
SCHOOL OF ENGINEERING
Storrs, Connecticut 06268

TECHNICAL REPORT UCT/DEECS/TR-86-3

ONR CONTRACT NUMBER N00014-85-K-0547

WORK UNIT NUMBER NR 662-006

March 1986

Department of
Electrical Engineering and Computer Science

"Carrier-Sense Stack Algorithms for
Multiple Access Communication Channels"

by

Lazaros Merakos

TECHNICAL REPORT UCT/DEECS/TR-86-3

ONR CONTRACT NUMBER N00014-85-K-0547

WORK UNIT NUMBER NR 662-006

March 1986

Principal Investigators
Prof. Panayota Papantoni-Kazakos
Prof. Lazaros Merakos

University of Connecticut
Electrical Engineering and Computer Science Department
U-157, 260 Glenbrook Rd.
Storrs, CT 06268

Distribution:
Scientific Officer
Administrative Contracting Officer
Director, Naval Research laboratory
Defense Technical Information Center

Carrier-Sense Stack Algorithms for Multiple Access Communication Channels

Lazaros Merakos

Electrical Engineering and Computer Science Department
University of Connecticut

Abstract

We consider the random multiple access of a collision-type, packet-switched channel, for the Poisson user model in a local area network environment, where "carrier sensing" techniques are possible due to small propagation delays. We propose and analyze random access algorithms that are representative of a new class of stable algorithms with "limited sensing" and "free access" characteristics. "Limited sensing" algorithms require that users sense the channel only while they have a packet to transmit, and, therefore, they have practical advantages over algorithms that require continuous channel sensing. The "free access" characteristics of the proposed algorithms simplify their implementation, since newly arrived packets are transmitted upon arrival, provided that the channel is sensed idle. Utilizing the regenerative character of the stochastic processes that are associated with the random access system, we derive lower bounds on the maximum stable throughput, and tight upper and lower bounds on the induced mean packet delay. The proposed algorithms are easy-to-implement, and they combine inherently stable operation and high performance with modest channel sensing requirements.

This work was supported by the Office of Naval Research under contract no. N00014-85-K-0547.

1. INTRODUCTION

Local area networks (LANs) are designed to support high bandwidth communications among a large number of users within a local geographical area. In the case of a large population of independent, bursty users this service can be provided at a low cost per user, if the LAN employs a single, packet-switched, collision-type, multiple access channel using a communications medium such as coaxial cable, optical fibre, or radio multi-access channel. The sharing of the common channel by the contending users is coordinated by a distributed control random access algorithm (RAA).

When the end-to-end propagation delay of the LAN is small, as compared to the transmission time of a packet, then the users can determine the channel activity in a short amount of time, through "channel sensing" operations. Depending on the communication medium used, users may be able to determine whether the channel is idle or busy,(i.e., the carrier-sensing environment), or even to discriminate between successful and interfering transmissions while they are in progress,(i.e., carrier-sensing with collision detection).

The earliest and most well known RAAs for the carrier-sensing environment belong to the class of ALOHA-type algorithms, such as the non-persistent CSMA and CSMA-CD algorithms and their variations,[1-3]. Ethernet,[4],is a prominent example of a LAN using an algorithm from this class. ALOHA-type algorithms are easy to implement, but they have inherent long-term stability problems, unless retransmission control algorithms are employed to support them,[5].

A relatively new class of random access algorithms is the class of Tree algorithms [6-10]. These algorithms gather information about the history of the channel activity (feedback information), and use it to resolve collisions by employing a tree-search-type collision resolution procedure. Tree-type algorithms, which are extensions of the Tree algorithms to the carrier-sensing environment, have been analyzed in [11,

12]. These algorithms have continuous channel sensing and blocked-access characteristics; that is, all users are required to inspect the feedback information by sensing the channel constantly even if they have nothing to transmit, and newly arrived packets are blocked until the on-going collision (if any) has been resolved. These algorithms are inherently stable, perform better than ALOHA-type algorithms, and some of them guarantee first come-first serve delivery of packets. However, the continuous channel sensing requirement, which is an integral part of their operation, makes the Tree-type algorithms unsuitable for networks where activation of new users and user mobility disrupt the feedback sensing continuity. In addition, some of the more efficient Tree-type algorithms are sensitive to errors in the feedback information caused by channel noise or by actions of higher level protocols discarding packets already in the system [10, 11].

A new trend towards the design of RAAs that could combine stable operation and high performance with modest feedback requirements and robustness in the presence of feedback errors started with the introduction of the "Stack" algorithm by Tsybakov and Vvedenskaya [12], and its variations [13-16]. The new class of algorithms has limited channel sensing and free-access characteristics. In contrast to Tree algorithms, the algorithms of this class require that a user sense the channel only while he has a packet to transmit (limited channel sensing); furthermore, newly arrived packets access the channel freely, independently of any collision resolution process that might be in progress. In addition to being practically appealing, the algorithms of this class are less sensitive to feedback errors, as compared to Tree algorithms, for the same channel and user model [16].

The above considerations have motivated our interest in limited channel sensing algorithms for the carrier-sensing environment. In this paper we propose and analyze a simple such algorithm, which is representative of the above class. Utilizing the regenerative character of the stochastic processes that are associated with the random

access channel, we derive lower bounds on the maximum stable throughput, and tight upper and lower bounds on the mean packet delay induced by the algorithm. These results indicate that the proposed algorithm has mean delay-throughput characteristics that are uniformly better than those of the optimally controlled non-persistent CSMA and CSMA-CD algorithm [5], and comparable to those induced by the extension of the most efficient Tree algorithm to the carrier-sensing environment [12]. More important, however, is the fact that the proposed algorithm combines high performance and inherently stable operation with limited channel sensing and low operational complexity.

The organization of the paper is as follows. Section 2 introduces the user and channel model.Section 3 states the algorithm. In section 4 we explain some of its important properties, and we evaluate its output rate. In section 5 we develop bounding techniques that yield arbitrarily tight upper and lower bounds on the mean packet delay induced by the algorithm, and a lower bound on the algorithm's maximum stable throughput. In section 6 we present a generalized version of the algorithm introduced in section 2, and we make some performance comparisons. Finally, in section 7 we draw some conclusions.

2. USER AND CHANNEL MODEL

We assume that an infinite population of independent, bursty, packet-transmitting users share a common communication channel. We model the packet arrival process as homogeneous Poisson with intensity λ packets per unit of time. For convenience, we assume that packets are of fixed length, and we take the packet transmission time to correspond to our unit of time. We also assume that the propagation delay between any two users in the network is at most α, where $\alpha < 1$.

For simplicity in analysis, we assume that the time axis is slotted, where the slot size is equal to the maximum propagation delay α. Users may initiate a packet transmission only at the beginning of a slot.

We consider limited channel sensing and ternary feedback. That is, each user senses the channel continuously, from the time instant when he generates a packet, to the time instant when this packet is successfully transmitted, and he can distinguish without error among the following channel states: a) idle (no transmission) b) success (transmission of a single packet) c) collision (simultaneous transmission of at least two packets). We assume that a collision results in complete loss of the information included in all the involved packets; thus, retransmission is then necessary.

Without loss of generality, we assume that a user who senses the channel can distinguish between transmission (success or collision) and no transmission (idle) instanteously.[1] However, the time required to distinguish a collision from a successful transmission (collision detect time) is a system characteristic whose value depends on the maximum propagation delay, the transmission medium, the packet encoding and modulation techniques, and the method used to detect collisions [21, 22].

In cable networks like Ethernet, because of the physical properties of the cable, it is possible for a user to listen to the cable while transmitting. What this means is that if more than one users start to transmit at the beginning of a slot, they will shortly determine that interference is in process and they will, subsequently, abort

---

[1] In channels where this sensing operation cannot be considered instantaneous, the slot size will be the sum of the maximum propagation delay and the time required by a receiver to reliably distinguish between transmission and no transmission.

their transmissions.

Given that a collision occurs, the time until all transmitting users stop transmission will be called the "conflict truncation time" and it will be denoted by β. The value of β depends on the implementation. In synchronous Ethernet-type networks, for example, the collision truncation time can be represented as the sum of three terms, $\beta = \gamma + \delta + \zeta$. The γ term represents the propagation delay before interference reaches all transmitting users; clearly, γ is less than or equal to the maximum propagation delay α. The δ term represents the time it takes for a user to determine interference once the latter has reached him. The ζ term denotes the time spent for a collision consensus reinforcement mechanism, by which a user, experiencing interference, jams the channel by transmitting additional bits, (usually, in the form of encoded phase violations), to ensure that all users who sense the channel detect the collision. The values of δ and ζ depend on the implementation,and can be as small as a few bits transmission time. Here we assume that $\alpha \leq \beta < 1$. In addition, we assume that the detection of collisions is performed by the receiver (receive mode collision detect, IEEE Standards Committee, Project 802, [23]). This means that, in addition to transmitting users, non-transmitting users have, also, the capability to detect collisions, provided that they are sensing the channel during the collision.

In contrast to the cable network users, the users is some local networks, such as packet radio networks, cannot listen to the channel while they are transmitting. If a collision occurs, then the transmitting users will detect the interference not earlier than the end of their transmission; thus, in this case the conflict truncation time is usually equal to the packet transmission time; i.e., $\beta = 1$.

In summary, the two important parameters of the carrier-sense channel considered here are the maximum propagation delay α, and the conflict truncation time β. The performance of the algorithms to be presented in this paper will be evaluated for values of α and β such that $a < 1$ and $\alpha \leq \beta \leq 1$. This range of parameters models

adequately well a substantial class of the cable and radio networks that use carrier-sense channels.

To facilitate the comparison of the performance of the algorithms to be presented here to that of other algorithms found in the literature, we assume that both the packet transmission time and the conflict truncation time are integer multiples of a slot; that is, we assume that the packet transmission time is equal to T slots, where $T = 1/\alpha > 1$, and that the conflict truncation time is equal to R slots, $1 \leq R \leq T$, where $R = \beta/\alpha$.

3. THE ALGORITHM AND ITS GENERAL OPERATION

In this section we describe a limited channel sensing algorithm that allows users to communicate with each other in a carrier-sensing environment satisfying the assumptions specified in the previous section.

The algorithm is implemented by each "busy" user in a distributed fashion. A user is defined to be busy from the moment it generates a new packet for transmission until the moment after the same packet is successfully transmitted; otherwise, the user is said to be idle. The time instant that a user generates a packet, (i.e., when he becomes busy), he starts sensing the channel and he simultaneously initializes the algorithm; he continues to sense the channel until the successful transmission of his packet, (i.e., until he becomes idle). Upon the occurrence of this event, he stops sensing the channel and simultaneously he terminates the algorithm.

For the implementation of the algorithm the user uses a counter, whose indication at time t is denoted by CI_t. The indications of the counter dictate the operation of the algorithm, which is described as follows:

Rule 1 -- Counter initialization

Let the user generate a new packet at time t_0, and let k_0 denote the first slot boundary, after t_0. Also, let k_1 denote the first slot boundary after t_0, at which the user senses the channel idle. Then, at k_1, the user initializes his counter as follows:

$$CI_{k_1} = \begin{cases} 1 \ ; \ \text{if } k_1 = k_0 \\ M \ ; \ \text{if } k_1 \neq k_0 \end{cases}$$

where M is a random variable uniformly distributed on $\{1,2,\ldots,m\}$, and the integer m, $m \geq 1$, is an algorithmic parameter.

Rule 2 -- Transmission rule

The user transmits at the beginning of the slots at which his counter indication equals "1".

Rule 3 -- Counter updating

After the user has initialized his counter he updates it only at the slot boundaries at which he senses the channel idle. Let $k_1, k_2, \ldots$ denote these slot boundaries in accordance with their occurrence. Let the user be busy at $k_i (i=1,2,3,\ldots)$, with $CI_{k_i} \geq 1$. Then, at time k_{i+1} he updates his counter as follows:

a) If $CI_{k_i} > 1$, then

$$CI_{k_{i+1}} = \begin{cases} CI_{k_i} - 1 & \text{if, during } (k_i, k_{i+1}) \text{, he senses the channel idle} \\ CI_{k_i} + m - 1 & \text{if, during } (k_i, k_{i+1}) \text{, he senses the channel busy with a successful transmission} \\ CI_{k_i} + m + n - 1 & \text{if, during } (k_i, k_{i+1}) \text{, he senses the channel busy with a collision} \end{cases}$$

where the integer $n, n \geq 2$, is an algorithmic parameter.

b) If $CI_{k_i} = 1$ and, during (k_i, k_{i+1}), he senses the channel busy with a collision, then

$$CI_{k_{i+1}} = m + J$$

where J is a random variable uniformly distributed on $\{1,2,\ldots,n\}$

If $CI_{k_i} = 1$ and, during (k_i, k_{i+1}), he senses the channel busy with a successful transmission, then his packet has been successfully transmitted and the user terminates the algorithm.

The integers m and n used in the description of the algorithm are design parameters subject to optimization for throughput maximization; their optimum values depend on the values of the system parameters α and β, and they will be given later.

The general operation of the algorithm is perhaps better illustrated by introducing the concept of a "stack". A stack is an abstract storage device consisting of an infinite number of cells, labelled 1,2,3,... . The number of packets that a cell can accommodate is unrestricted. At each time t during the operation of the algorithm, users with counter value $CI_t = r$ can be thought of as having stored their

packets in cell #r of the stack. A packet is transmitted whenever it enters cell #1 of the stack. Packets are, eventually, successfully transmitted after moving through the cells of the stack in accordance with the algorithmic rules described above.

The execution of the algorithm by each busy user induces on the time axis an alternate sequence of transmission periods (successful or unsuccessful) and idle periods. However, this channel activity reaches each of the users, who sense the channel, with a different amount of delay, depending on their distance from the transmitting users. For convenience, consider an arbitrary user, called user X, and assume that he senses the channel continuously from the beginning of the operation of the system. Let $t_i (i=0,1,2,...)$ denote the consequtive slot boundaries at which user X senses the channel idle. The interval $[t_i, t_{i+1})$, $i=0,1,2,...$, will be called the i-th algorithm step. If during an algorithm step the channel is idle, busy with a successful transmission, or busy with a collision then the algorithm step will be called idle, successful, or unsuccessful, respectively. As it can be seen from figure 1 an idle algorithm step lasts for one slot; a successful algorithm step lasts for T+1 slots, T slots to place the packet onto the channel and one slot for this packet to clear the channel due to propagation delay; an unsuccessful algorithm step lasts for R+1 slots, R slots for the transmitting users to detect the collision and abort their transmissions and one slot for the packet fragments to clear the channel due to propagation delay. Thus, the length of the i-th algorithm step, measured in units of time, is given by

$$t_{i+1} - t_i = \begin{cases} \alpha & \text{if the i-th algorithm step is idle} \\ 1+\alpha & \text{if the i-th algorithm step is successful} \\ \alpha+\beta & \text{if the i-th algorithm step is unsuccessful} \end{cases}$$

The description of the general operation of the algorithm and its analysis are greatly facilitated if one considers how the state of the stack evolves at the beginning of consequtive algorithm steps. In figure 2 the stack is imbedded at t_i

and t_{i+1} to show how packets move through the cells of the stack,(i.e.,how users update their counters),as well as to show how new packets arriving between t_i and t_{i+1} place themselves in the cells of the stack,(i.e.,how users initialize their counters), depending on whether the algorithm step was idle, successful or unsuccessful.

As it can been seen from figure 2, the operation of the algorithm is based on the "divide and conquer" philosophy that characterizes most RAAs. More specifically, the algorithm spreads the incoming traffic into the first m cells of the stack to, a priori, avoid collisions, when the new traffic is heavy,(e.g., after a successful transmission). Furthermore, to resolve collisions, it uniformly splits the group of collided packets into n cells of the stack. The parameters m and n allow the algorithm to adapt its operation to the given values of the network parameters a and β. If, for example, $\alpha << \beta$, then m and n should be large to take advantage of the much lower "cost" (wasted channel time) of an idle algorithm step (α units of time), as compared to that of an unsuccessful algorithm step ($\beta+\alpha$ units of time). Finally, we point out that users with newly arrived packets initialize their counters only on the basis of whether the channel is busy or idle (see rule 1). This is desirable, since some of them may not have sufficient time to reliably distinguish a successful transmission from a collision , before the channel goes idle.

The algorithm described in this section will be referred to as the LAN stack algorithm (LANSA).

4. RENEWAL PROPERTIES AND OUTPUT RATE

To analyze the performance of the LANSA we introduce the concept of a session. A session is a sequence of consecutive algorithm steps that begins and ends at two consecutive algorithm renewal instants. These instants are denoted by R_n, $n \geq 1$, and are determined by means of a conceptual marker that operates on the stack. The first session begins with the beginning of the first algorithm step, at $R_1 = t_1$, with the marker placed at cell #2. During the session, the marker's position in the stack is adjusted at the beginning of each algorithm step. At t_i, let the marker be at cell $\#C_i$, $C_i \geq 2$; then, at t_{i+1} the marker is placed at cell $\#C_{i+1}$, with

$$C_{i+1} = \begin{cases} C_i - 1 & \text{if the ith algorithm step is idle} \\ C_i + m - 1 & \text{if the ith algorithm step is successful} \\ C_i + m + n - 1 & \text{if the ith algorithm step is unsuccessful} \end{cases}$$

where the integers $m \geq 1$, $n \geq 2$ are as defined in the LANSA description.

The second renewal instant, R_2, is the instant at which the marker drops to cell #1 for the first time, that is, $R_2 = \min \{t_i > R_1 : C_i = 1\}$; this signifies the end of the first session. Instantaneously, at R_2, the marker is then adjusted to cell #2 and the second session begins. This process continues indefinitely.

A session starting with k packets in the first cell of the stack is called a session of multiplicity k, $k \geq 0$. Note that if at t_i, $i \geq 1$, the marker is at cell #r of the stack then, from the rules of the algorithm and the marker's instructions, it is deduced that cells #j, $j \geq r$, are necessarily empty. Thus, when a session begins, all cells are empty except for cell #1, which is occupied by the k new packets that arrived during the last algorithm step of the previous session.

The time from the instant that a session begins until it ends is the length of the session. The session with multiplicity 0 is called the empty session, and has length equal to α, (i.e., one slot). A non empty session has a random length that depends on the arrival process of new packets during the session, and on the rules

of the algorithm. In view of the independent and stationary increments property of the Poisson process that models the input traffic, it follows that the session lengths will be independent, identically distributed (i.i.d.) random variables, if the session multiplicities are i.i.d. random variables. Note that a session ends when the marker drops to cell #1 for the first time. Then, since the marker's position is decremented only after an idle algorithm step (see marker's instructions), it follows that the last algorithm step of a session is always idle. Thus, the session multiplicities are i.i.d. random variables with distribution

$$P(K=k) = p_k \triangleq (\lambda\alpha)^k \exp(-\lambda\alpha)/k!, \tag{1}$$

and, therefore, the session lengths are i.i.d. random variables as well.

a. Output Rate

Let L_1, L_2, ... denote the lengths of successive sessions; then,

$$R_1 = \alpha \; ; \; R_{i+1} = R_i + L_i \;, \; i=1,2,...$$

define the algorithm renewal instants. The sequence $\{R_i\}_{i\geq 1}$ forms a delayed renewal process, since L_1, L_2, ... are i.i.d. non-negative random variables.

Let

$$\mu(n) = t_n^{-1} \sum_{j=1}^{n} I(O_j)$$

where $I(O_j)$ denotes the indicator function of the event $O_j \triangleq$ {successful transmission during the jth algorithm step}. Thus, $\mu(n)$ represents the average fraction of time that successful packet transmissions have occurred on the channel by instant t_n.

Consider now an arbitrary session, say the ith, and let S_i denote the random number of packets that were successfully transmitted during the course of the session. Clearly, S_i depends on L_i, but the pairs (L_i, S_i), $i \geq 1$, are independent and identically distributed. Let $S = E(S_i)$, and $L = E(L_i)$. The expected number of

successful transmissions, S, during a session can be thought of as an average reward earned during the session. With this in mind, we state the following result from the theory of renewal reward processes (see, for example, [18, sec. 3.6]).

Theorem 1 If $S < \infty$, and $L < \infty$, then there exists a real number, μ, such that

$$\lim_{n\to\infty} \mu(n) = \lim_{n\to\infty} E(\mu(n)) = \frac{S}{L} = \mu \quad \text{with probability 1}$$

The above theorem states that the (expected) long-run average number of successful transmissions per unit time is just the expected number of successful transmissions during a session, divided by the mean session length, provided that both S and L are finite. The quantity μ is the channel's output rate.

Consider again the ith session; since sessions always end with an idle algorithm step, and at both R_i and R_{i+1} there are no blocked users present in the system, it follows that the number, S_i, of successfully transmitted packets (if any) during the ith session is just the number of packets arrived at the system during the time interval $[R_i-\alpha,\ R_{i+1}-\alpha)$; all such packets will be called packets associated with session i. Now, if we let $M_i \triangleq L_i/\alpha = (R_{i+1}-R_i)/\alpha$ denote the length of the ith session measured in slots, then

$$S_i = \sum_{j=1}^{M_i} A_j \tag{2}$$

where A_j denotes the number of arrivals in the interval $[R_n+(j-2)\alpha,\ R_n+(j-1)\alpha)$ Clearly, $\{A_j\}_{j\geq 1}$ is a sequence of independent Poisson random variables, with intensity $\alpha\lambda$. Furthermore, M_i is a stopping time for $\{A_j\}_{j\geq 1}$ since the event $\{M_i=m\}$ is independent of A_{m+1}, A_{m+2}, ... If we assume that $E(M_i) < \infty$, then, taking expectations in (2), and applying Wald's lemma, ([17], p. 59) yields

$$S = E(S_i) = E(M_i)E(A_j) = \lambda\ L \tag{3}$$

since $E(A_j) = \alpha\lambda$, and $E(M_i) = L/\alpha$, from the definition of M_i. In view of (3), theorem 1 yields

Corollary 1. If $L < \infty$, then $\mu = \lambda$; that is, the LANSA maintains the rate if the mean session length is finite.

In the rest of this section we investigate the conditions under which $L < \infty$, and we establish bounds on L.

b. Mean Session Length

Consider a session of random multiplicity, $K \geq 0$, and let ℓ_K denote its random length. If we let $L_k = E(\ell_K | K=k)$, then the mean session length is given by

$$L = \sum_{k=0}^{\infty} p_k L_k \tag{4}$$

where p_k is as given by (1).

We proceed now with the investigation of the region of convergence of the series given in (4), by deriving and studying a system of equations for the mean length of a session of specified multiplicity, L_k, $k \geq 0$. We first state the following.

Proposition 1. The length, ℓ_k, of a session with specified multiplicity $k \geq 0$ satisfies the following system of equations:

$$\ell_k = \begin{cases} \alpha & \text{if } k = 0 \\ 1+\alpha+\sum_{j=1}^{m} \ell_{X_j} & \text{if } k = 1 \\ \beta+\alpha+\sum_{j=1}^{m} \ell_{Y_j} + \sum_{j=1}^{n} \ell_{K_j} + Z_j & \text{if } k \geq 2 \end{cases} \tag{5}$$

where $X_1,\ldots,X_m$, $Y_1,\ldots,Y_m$, $Z_1,\ldots,Z_n$ are independent random variables, which are, also, independent of the random variables $K_1,\ldots,K_n$. The corresponding distributions are as follows: $P(X_j=i) = s_i \overset{\Delta}{=} (\lambda_s)^i \exp(-\lambda_s)/(i!)$, $P(Y_j=i) = r_i \overset{\Delta}{=} (\lambda_r)^i \exp(-\lambda_r)/(i!)$, $1 \leq j \leq m$, where $\lambda_s = (m^{-1}+\alpha)\lambda$, and $\lambda_r = (\beta m^{-1}+\alpha)\lambda$; $P(Z_j=i) = p_i$, as defined in (1), $1 \leq j \leq n$;

$$P(K_1=k_1,\ldots,K_n=k_n) = \frac{k!}{k_1!k_2!\ldots k_n!}\; n^{-k}\;,\; 0 \le k_i \le k\;;\; \sum_{i=1}^{n} k_i = k.$$

Proposition 1 follows directly from the LANSA specifications, and its proof is omitted. Taking expectations in (5) yields the following proposition.

Proposition 2. The mean session lengths, $\{L_k\}_{k\geq 0}$, satisfy the following infinite dimensional system of linear equations:

$$x_0 = \alpha \quad ; \qquad x_k = \sum_{i=1}^{\infty} a_{k,i}\, x_i + g_k\;,\; k \geq 1 \tag{6}$$

where

$a_{1,i}=ms_i$; $a_{k,i}=mr_i + nq_{k,i}$, $k>1$; $g_1=1+(1+ms_0)\alpha$; $g_k = \beta+(1+mr_0+nq_{k,0})$, $k>1$;

where[1]

$q_{k,i}\triangleq p_i * b_i(k,n^{-1})$, $b_i(k,p) \triangleq \binom{k}{i} p^i(1-p)^{k-i}$, and s_i, r_i, p_i are as defined in proposition 1.

Formally, the system of equations (6) always has an "infinite" solution $x_0 = \alpha$, $x_k = \infty$, $k \geq 1$. The following theorem specifies a sufficient condition under which system (6) has a solution, $\{x_k\}_{k\geq 0}$, with $0 \leq x_k < +\infty$ for all $0 \leq k < \infty$, which coincides with the sequence, $\{L_k\}_{k\geq 0}$, of the mean session lengths induced by the LANSA.

Theorem 2

(i) Given $\alpha,\beta,m\geq 1$, $n\geq 2$, system (6) has a solution, $\{y_k\}_{k\geq 0}$, such that

$$y_0 = \alpha,\; 0 < b'k-c' \leq y_k \leq bk-c,\; k \geq 1$$

if $\lambda < \lambda_o(\alpha,\beta;m,n)$; where $\lambda_o(\alpha,\beta;m,n)$ is the unique solution of the equation

---

1. * denotes convolution

$$\frac{1-(1+m\alpha)\lambda}{1-m(1-s_0)} - \lambda \frac{\beta+(m+n)\alpha}{(1-r_0)m+(1-q_{2,0})n-1} = 0 \tag{7}$$

over the interval $[0,(1+m\alpha)^{-1})$, and where the coefficients b,b',c, and c' are bounded functions of λ.

(ii) For every $\lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n))$, $L_k = y_k$, for all $k \geq 0$.

The proof of theorem 2, and the expressions for the coefficients b,b',c, and c' can be found in the Appendix.

Given α,β,m,n, let $\lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n))$; then, from theorem 2, we have

$$b'\ k-c' \leq L_k \leq b\ k-c\ ,\ k \geq 1 \tag{8}$$

Substitution of the above bounds into (4) yields

$$L_\ell \leq L \leq L_u < \infty\ ,\ \text{for every}\ \lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n))\ , \tag{9}$$

where $L_u = \alpha\lambda b-c + (\alpha+c)\exp(-\alpha\lambda)$, $L_\ell = \alpha\lambda b'-c' + (\alpha+c')\exp(-\alpha\lambda)$

Given the network parameters α,β, let us now define,

$$\bar{\lambda}(\alpha,\beta) \overset{\Delta}{=} \sup_{m,n}\{\lambda_o(\alpha,\beta;m,n)\} = \lambda_o(\alpha,\beta;m^*,n^*) \tag{10}$$

In view of (9), (10), and of corollary 1 we have the following corollary.

<u>Corollary 2</u>. Given α,β,m,n, the LANSA maintains the rate, that is, $\lambda = \mu$, for every $\lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n))$. Given α,β, the LANSA with $m = m^*$ and $n = n^*$ maintains the rate for every $\lambda\varepsilon[0,\bar{\lambda}(\alpha,\beta))$.

We used numerical search techniques to determine $\bar{\lambda}(\alpha,\beta)$. In table 1 we give the values of $\bar{\lambda}(\alpha,\beta)$ for representative values of α, and for β=α, β=0.5, and β=1. In the same table we also give the values, m^*, n^*, of the design parameters m,n that achieve the maximization in (10). As table 1 reveals, m^* and n^* generally increase as α

decreases; this is more prominent in the case where $\beta >> \alpha$ (bottom left part of the table). This is an intuitively pleasing result; if the cost of an unsuccessful algorithm step is much higher than the cost of an idle algorithm step, then, to avoid collisions, the algorithm should a priori spread the incoming traffic by increasing m, and it should split the group of collided packets into more cells by increasing n.

c. Tight Bounds on the Mean Session Length

The bounds on L given by (9) are tight enough for small values of λ, but they become loose as λ approaches $\lambda_o(\alpha,\beta;m,n)$. In this subsection we develop a method for computing bounds on L that are tighter than those given by (9); these new bounds will be used in the delay analysis of the next section.

Given some finite natural number, $N \geq 1$, let us consider the following system of N linear equations

$$x_k = \sum_{i=1}^{N} a_{k,i}\, x_i + b_k\ , \quad 1 \leq k \leq N, \qquad (11)$$

where b_k, $1 \leq k \leq N$, are non-negative real constants, and $a_{k,i}$, $1 \leq i \leq N$, $1 \leq k \leq N$, are as given in proposition 2.

The solution to system (11) is characterized by the following lemma, whose proof can be found in the Appendix.

Lemma 1

Given α,β,m,n, let $\lambda < \lambda_o(\alpha,\beta;m,n)$; then, for every $N \geq 1$ and for every given $b_k \geq 0$, $1 \leq k \leq N$, system (11) has a unique non-negative solution $\underline{x} = (I_N-A_N)^{-1}\underline{b}$, and the matrix $(I_N-A_N)^{-1}$ has non-negative elements; where[2] $\underline{x} = (x_1,\ldots,x_N)^t$, $\underline{b} = (b_1,\ldots,b_N)^t$, $A_N = (a_{ij})$ is the (NxN) non-negative, square matrix with $a_{ij} = a_{i,j}$, $1 \leq i \leq N$, $1 \leq j \leq N$, and I_N is the (NxN) identity matrix.

Using lemma 1 we can express the following theorem, whose proof is given in the Appendix.

2. t denotes transpose

Theorem 3

Given α,β,m,n, let $\lambda < \lambda_o(\alpha,\beta;m,n)$; then, for every $N \geq 1$

$$b'k-c' \leq L_k^{\ell} \leq L_k \leq L_k^u \leq bk-c \ , \ 1 \leq k \leq N, \tag{12}$$

where $\{L_k^u;\ 1\leq k\leq N\}$, and $\{L_k;\ 1\leq k\leq N\}$ are the unique solutions to system (11) with $b_k = g_k + \sum_{i=N+1}^{\infty} a_{k,i}(bi-c)$, and $b_k = g_k + \sum_{i=N+1}^{\infty} a_{k,i}(b'i-c')$, respectively.

Using the bounds L_k^u, L_k^{ℓ}, for $1 \leq k \leq N$, and the linear bounds given by (8), for $k > N$, in (4) we have

$$L_{\ell}^* \leq L \leq L_u^* \ , \ \text{for every} \ \ \lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n)) \tag{13}$$

where

$$L_u^* = p_0 a + \sum_{k=1}^{N} p_k L_k^u + \sum_{k=N+1}^{\infty} p_k(bk-c) = L_u - \sum_{k=1}^{N} p_k(bk-c-L_k^u) \tag{14}$$

and

$$L_{\ell}^* = p_0 a + \sum_{k=1}^{N} p_k L_k^{\ell} + \sum_{k=N+1}^{\infty} p_k(bk-c) = L_{\ell} + \sum_{k=1}^{N} p_k(L_k^{\ell}-b'k-c') \tag{15}$$

and where L_u, L_{ℓ} are as given in (9). Note that, from (12), (14), and (15), we have $L_{\ell} \leq L_{\ell}^* \leq L_u^* \leq L_u$. The bounds $\{L_k^u;\ 1\leq k\leq N\}$, and $\{L_k^{\ell};\ 1\leq k\leq N\}$, required for the evaluation of L_u^*, and L_{ℓ}^*, can be obtained by solving finite system (11), with b_k, $1\leq k\leq N$, as defined in theorem 3. Using N = 10, we solved system (11), for several representative values of the network parameters α,β, and for $m=m^*$, $n=n^*$, as given in table 1. The results for $(\alpha,\beta)=(.1,.1)$ and $(\alpha,\beta)=(.1,1)$ are included in table 2. In table 3, we give the bounds L_u^*, and L_{ℓ}^*, as found by substituting the solutions $\{L_k^u;\ 1\leq k\leq 10\}$, and $\{L_k^{\ell};\ 1\leq k\leq 10\}$ into (14), and (15), respectively. Note that the bounds L_u^*, and L_{ℓ}^* remain extremely tight, (they coincide up to at least the sixth decimal point), even for λ very close to $\bar{\lambda}(\alpha,\beta)$. We should also point out that, by increasing the dimensionality N of the finite system (11), arbitrarily tight bounds can be obtained.

5. DELAY ANALYSIS AND STABILITY

Consider the network operating with the LANSA over the time interval $[0, +\infty)$. Packets arrive at the network at time instants a_n, n=1,2,..., where $0 \le a_1 \le a_2 \le \ldots$. Let the arriving packets be labelled n=1,2,... according to their arrival instant. We define the delay, $\mathcal{D}_n$, experienced by the nth packet as the time difference between its arrival at the transmitter and the instant it is successfully received by the most remote receiver[3], (so that $\mathcal{D}_n = 1+\alpha$, when the packet is successfully transmitted beginning at the same moment it arrives at the transmitter). Let the random variable N_i denote the total number of packets associated with, (i.e., arrived and successfully transmitted during) the first i non-empty sessions. Let, also, $\bar{S}_i$ denote the number of packets associated with the ith non-empty session. We have that $N_0=0$, $N_{i+1}=N_i+\bar{S}_{i+1}$, i=0,1,2,... . The sequence $\{N_i\}_{i\ge 0}$ is a renewal process, since $\{\bar{S}_i\}_{i\ge 1}$ is a sequence of i.i.d. random variables. Furthermore, the renewal properties of the LANSA clearly indicate that, whenever a non-empty session ends, the arrival and waiting-time mechanisms are "reset" by the next arrival; thus, the process $\{\mathcal{D}_{N_i+n}\}_{n\ge 1}$, for every $i \ge 0$, is a probabilistic replica of the process $\{\mathcal{D}_n\}_{n\ge 1}$. Thus, the discrete-time process $\{\mathcal{D}_n\}_{n\ge 1}$ is <u>regenerative</u> respective to the imbedded renewal process $\{N_i\}_{i\ge 0}$, with common regenerative cycle, $\bar{S}$, the number of packets associated with a non-empty session.

Next define $\bar{S} = E(\bar{S})$, and $\bar{T} = E\left(\sum_{i=1}^{\bar{S}} \mathcal{D}_i\right)$; note that $\bar{T}$ represents the mean cumulative delay experienced by all the packets of a non-empty session. Using $\bar{S}$ and and $\bar{T}$ we can express the following standard result from the theory of regenerative processes (see [19, Thm. 2], and [20, Thm. 37]).

<u>Theorem 4</u>

If (A.1) $\bar{S}$ is not periodic, with $\bar{S} < \infty$ and if (A.2) $\bar{T} < \infty$, then there exists a real number D such that

3. This is the worst case, since the propagation delay from the transmitter to the intended receiver is at most α.

$$D = \ell im_{n\to\infty} n^{-1} \sum_{i=1}^{n} \mathcal{D}_i = \ell im_{n\to\infty} n^{-1} E\left(\sum_{i=1}^{n} \mathcal{D}_i\right) \quad \text{with probability 1.}$$

Furthermore, $\mathcal{D}_n$ converges in distribution to a random variable $\mathcal{D}_\infty$, and

$$D = E(\mathcal{D}_\infty) = \bar{T}/\bar{S} < \infty$$

Thus, under assumptions (A.1) and (A.2), the limiting average, the limiting expected average, and the mean of the limiting distribution of $\{\mathcal{D}_n\}_{n\geq 1}$ exist, coincide, and are finite; their common value, D, represents the mean packet delay induced by the LANSA. Next we elaborate on the assumptions stated in theorem 3. From the operation of the LANSA it can be easily seen that $\bar{S}$ is not periodic. Let S_k denote the expected number of packets associated with a session of multiplicity $k \geq 0$. Then, noting that $S_0 = 0$, we have

$$\bar{S} = \frac{1}{1-p_0} \sum_{k=1}^{\infty} p_k S_k = \frac{1}{1-p_0} \sum_{k=0}^{\infty} p_k S_k = \frac{1}{1-p_0} S \tag{16}$$

Given α,β,m,n, let $\lambda\epsilon[0, \lambda_o(\alpha,\beta,m,n))$; then, from (3), (16), and theorem 3, it follows that

$$\bar{S} = (1-p_0)^{-1} \lambda L < \infty \tag{17}$$

Thus, assumption (A.1) is true, if $\lambda < \lambda_o(\alpha,\beta;m,n)$. We proceed now to show that this is also true for assumption (A.2).

Consider the nth packet arrival; its delay can be expressed as

$$\mathcal{D}_n = A_n + C_n$$

A_n denotes the nth packet access waiting time from the packet arrival instant to the instant the packet enters the stack for the first time.

C_n denotes the nth packet contention waiting time from the instant the packet enters the stack for the first time to the instant it is successfully received by the most remote receiver.

Using the above decomposition of the packet delay, we write the mean cumulative delay, $\bar{T}$, experienced by all packets associated with a non-empty session as

$$\bar{T} = U + V \tag{18}$$

where $U \overset{\Delta}{=} E(\sum_{i=1}^{\bar{S}} A_i)$, and $V \overset{\Delta}{=} E(\sum_{i=1}^{\bar{S}} C_i)$ represent the mean cumulative access delay and the mean cumulative contention delay, respectively, over a non-empty session. Consider the access waiting time, A_i, of the ith packet of a non-empty session. Rule 1 of the algorithm implies that

$$A_i \leq \begin{cases} \alpha & \text{if the ith packet arrived during an idle algorithm step} \\ \beta+\alpha & \text{if the ith packet arrived during an unsuccessful algorithm step} \\ 1+\alpha & \text{if the ith packet arrived during a successful algorithm step} \end{cases}$$

Clearly then $A_i \leq \max(\alpha,\beta+\alpha,1+\alpha) = 1+\alpha$. Thus, $U \leq (1+\alpha)\bar{S}$, and in view of (17) we have that

$$U < \infty, \text{ for every } \lambda < \lambda_o(\alpha,\beta;m,n) \tag{19}$$

Next we consider the mean cumulative contention delay, V, experienced by all packets associated with a non-empty session. Let V_k denote the cumulative contention delay over a session of multiplicity $k \geq 0$. The rules of the algorithm yield the following relation for V_k:

<u>Proposition 3</u>

$$V_k = \begin{cases} 0 & k = 0 \\ 1+\alpha+\sum_{j=1}^{m} \ell_{X_j} + \sum_{j=1}^{m-1} \bar{X}_j \ell_{X_j} & k = 1 \\ (\beta+\alpha)k+(\sum_{j=1}^{m} \ell_{Y_j})k + \sum_{i=1}^{n-1} \bar{K}_i \ell_{K_i} + Z_i + & \\ + \sum_{j=1}^{m-1} \bar{Y}_j \ell_{Y_j} + \sum_{j=1}^{m} V_{Y_j} + \sum_{i=1}^{n} V_{K_i+Z_i} & k \geq 2 \end{cases} \tag{20}$$

where

X_j, Y_j, $1 \leq j \leq m$, and K_i, $1 \leq i \leq n$ are as given in proposition 1;

$\bar{K}_i = \sum_{j=i+1}^{n} K_j$, $1 \leq i \leq n-1$;

$\bar{X}_j$ ~ Poisson with intensity $(1-j/m)\lambda$, and $\bar{X}_j$ is independent of X_j, $1 \leq j \leq m-1$;

$\bar{Y}_j$ ~ Poisson with intensity $(1-j/m)\beta\lambda$, and $\bar{Y}_j$ is independent of Y_j, $1 \leq j \leq m-1$.

The proof is straightforward and will be omitted. We note only that, for $k = 1$, the first sum represents the total cumulative contention delay experienced by the packets of m independent subsessions with appropriate multiplicities, while the second sum represents the cumulative waiting time before first transmission of all the packets that were initially placed in cell #j, where $1 < j \leq m$. Similarly, for $k \geq 2$, the first two sums represent the cumulative delay experienced by the k collided packets while waiting for their first retransmission; the third sum represents the total waiting time before first transmission of all the packets that arrived during the collision of the k packets, and were intially placed in cell #j, where $1 < j \leq m$; the fourth and fifth sum represent the total cumulative contention delay associated with m and n subsessions, respectively, with appropriate multiplicities.

Taking expectations in both sides of (20) yields the following

Proposition 4. The mean cumulative contention delay, $V_k = E(\mathcal{V}_k)$, experienced by all packets associated with a session of multiplicity $k \geq 0$, satisfies the following system of equations

$$x_0 = 0 \; ; \; x_k = \sum_{i=1}^{\infty} a_{k,i} \, x_i + f_k \; , \; k \geq 1 \tag{21}$$

where

$$f_1 = f_1(\{L_i\}_{i\geq 0}) = 1+\alpha+\frac{1}{2}\lambda(m-1)\sum_{i=0}^{\infty} s_i\, L_i,\quad f_k = f_k(\{L_i\}_{i\geq 0}) = (\beta+\alpha)k +$$

$$+(km+\frac{1}{2}\beta\lambda(m-1))\sum_{i=0}^{\infty} r_i L_i + \sum_{\xi=0}^{n-2}\sum_{j=0}^{k}\sum_{\ell=0}^{\infty}\sum_{i=0}^{k-j} b_j(k,\frac{\xi}{n})b_i(k-j,\frac{1}{n-\xi})p_\ell(k-j-i)L_{i+\ell},\; k \geq 1,$$

where p_i, s_i, r_i, and $b_i(\cdot,\cdot)$ are as defined in propositions 1 and 2.

Note that system (21) differs from system (6) only in the forcing terms f_k. The following is a result analogous to theorem 2.

Theorem 5

Given $\alpha,\beta,m \geq 1$, and $n \geq 2$, let $\lambda_o(\alpha,\beta;m,n)$ be as defined in theorem 2. Then, for every $\lambda\epsilon[0, \lambda_o(\alpha,\beta;m,n))$,

(i) system (21) has a solution, $\{z_k\}_{k\geq 0}$, such that

$$z_0 = 0;\; 0 \leq \ell_1 k^2 + \ell_2 k + \ell_3 \leq z_k \leq u_1 k^2 + u_2 k + u_3\;,\; k \geq 1, \tag{22}$$

where the coefficients ℓ_1, ℓ_2, ℓ_3, u_1, u_2, and u_3 are bounded functions of λ.

(ii) the mean cumulative contention delays, $\{V_k\}_{k\geq 0}$, coincide with the solution $\{z_k\}_{k\geq 0}$, that is, $V_k = z_k$, for all $k \geq 0$.

The coefficients ℓ_1, ℓ_2, ℓ_3, u_1, u_2, and u_3 are derived in the proof of the theorem, which can be found in the Appendix.

Next we write the mean cumulative contention delay, V, over a nonempty session as

$$V = (1-p_0)^{-1}\sum_{k=1}^{\infty} p_k\, V_k \tag{23}$$

From theorem 5 we have that, for every $\lambda\epsilon[0, \lambda_o(\alpha,\beta;m,n))$,

$$0 \le \ell_1 k^2 + \ell_2 k + \ell_3 \le V_k \le u_1 k^2 + u_2 k + u_3 \ , \ k \ge 1 \tag{24}$$

Substitution of the above bounds into (23) yields

$$V_\ell \le V \le V_u < \infty \ , \text{ for every } \lambda \varepsilon [0, \ \lambda_o(\alpha,\beta;m,n)) \ , \tag{25}$$

where

$$V_u = (\alpha\lambda(1+\alpha\lambda)u_1+\alpha\lambda u_2+(1-e^{-\alpha\lambda})u_3)/(1-e^{-\alpha\lambda})$$

$$V_\ell = (\alpha\lambda(1+\alpha\lambda)\ell_1+\alpha\lambda\ell_2+(1-e^{-\alpha\lambda})\ell_3)/(1-e^{-\alpha\lambda})$$

From (25), and in view of (17), (18), and (19) we have that, for every $\lambda \varepsilon [0, \lambda_o(\alpha,\beta;m,n))$ both assumption (A.1) and assumption (A.2) in theorem 4 are true; thus, theorem 4 yields the following corollary.

Corollary 3. The mean packet delay, $D(\lambda)$, is finite for every $\lambda \varepsilon [0, \lambda_o(\alpha,\beta;m,n)$.

a. Stability

A random-access algorithm is called stable if the mean packet delay is finite. The maximum stable throughput, η, of a random-access algorithm is defined as the supremum of the cumulative input rate, λ, so that the algorithm is stable; that is, $\eta \triangleq \sup\{\lambda : D(\lambda) < \infty\}$.

Given the network parameters α,β, let $\eta(\alpha,\beta)$ denote the maximum stable throughput of the LANSA; then, from (10) and corollary 3, we have

$$\bar{\lambda}(\alpha,\beta) \le \eta(\alpha,\beta)$$

since $\lambda < \lambda_o(\alpha,\beta,m,n)$ is only a sufficient condition for finite mean packet delay. A plot of the lower bound $\bar{\lambda}$ on the maximum stable throughput induced by the LANSA, as a function of α, for $\beta=\alpha$ and $\beta=1$, is presented in figure 3. In the same figure we show throughput comparisons between the LANSA and the optimally controlled non-persistent CSMA and CSMA/CD algorithms presented in [5]. These algorithms assume that the users are aware of, or can estimate the number of blocked packets currently

in the system; this additional information is, then, used to implement retransmission control policies that stabilize the unstable non-persistent CSMA and CSMA/CD. As it can be seen from figure 3, the LANSA throughput is uniformly greater than that induced by the CSMA algorithms. Note, also, that as the propagation delay, α, increases, the throughput differences between the LANSA and the CSMA algorithms become significant.

In figure 4 we plot $\bar{\lambda}$ as a function of the packet lentgh T (in slots) for representative values of the conflict truncation time R (in slots). The effect of the early collision detection is perhaps better illustrated in figure 5, where $\bar{\lambda}$ is plotted as a function of the conflict truncation time R, with T as a parameter. For each T, the left end point of the curve corresponds to R = 1, and the right end point to R = T.

b. Bounds on the Mean Packet Delay

Given α,β,m,n, and $\lambda < \lambda_o(\alpha,\beta,m,n)$, then from theorem 4 and (18) we have

$$D = \bar{T}/\bar{S} = A + C$$

where $A = U/\bar{S}$, and $C = V/\bar{S}$ are the mean access delay and the mean contention delay, respectively.

We consider first the mean access delay. The process $\{A_n\}_{n\geq 1}$ is regenerative with respect to the imbedded renewal process $\{N_i\}_{i\geq 0}$, defined at the beginning of this section, for the same reason that $\{\mathcal{D}_n\}_{n\geq 1}$ is. Thus, theorem 3 applies to $\{A_n\}_{n\geq 1}$ as well. In particular, we have that A_n converges in distribution to a random variable A_∞ with $A = E(A_\infty) = U/\bar{S}$. Now, Since $E(A_n) \leq 1 + \alpha$, it follows that A_n is uniformly integrable. Thus,

$$\lim_{n\to\infty} E(A_n) = E(A_\infty) = A \tag{26}$$

Next we define the process $\{Z(t),\ t \geq 0\}$ as follows

$$Z(t) = \begin{cases} 0 & \text{if } t\epsilon[t_{n-1},t_n) \text{ and the nth algorithm step is idle} \\ 1 & \text{if } t\epsilon[t_{n-1},t_n) \text{ and the nth algorithm step is successful} \\ 2 & \text{if } t\epsilon[t_{n-1},t_n) \text{ and the nth algorithm step is unsuccessful} \end{cases}$$

Let a_n be the arrival instant of the nth packet. Using the process Z(t) we write

$$E(A_n) = E(E(A_n|Z(a_n))) \tag{27}$$

Now, since the arrival process is Poisson we have that the conditional distribution of A_n conditioned on the events $\{Z(a_n)=j\}$, for j=0,1, or 2, is uniform over an interval of length α, $1 + \alpha$, or $\beta+\alpha$, respectively. Thus,

$$E(A_n|Z(a_n)) = \frac{\alpha}{2} I(Z(a_n) = 0) + \frac{1+\alpha}{2} I(Z(a_n) = 1) + \frac{\beta+\alpha}{2} I(Z(a_n) = 2) \tag{28}$$

where $I(\cdot)$ is the indicator function of the event in the parenthesis.

From (27) and (28) we have

$$E(A_n) = \frac{\alpha}{2} P(Z(a_n) = 0) + \frac{1+\alpha}{2} P(Z(a_n) = 1) + \frac{\beta+\alpha}{2} P(Z(a_n) = 1) \tag{29}$$

Next we give a result relating to the asymptotic behavior of the process Z(t).

<u>Lemma 2</u> Given α,β,m,n, let $\lambda < \lambda_o(\alpha,\beta;m,n)$; then

$$\lim_{t\to\infty} P(Z(t)=j) = \lim_{n\to\infty} P(Z(a_n)=j)=\pi_j \;;\; j = 0,1,2$$

where

$$\pi_0 = \alpha(m-1)\lambda+\alpha(m+n-1)(1-1+m\alpha)\lambda)(\beta+\alpha(m+n)^{-1}+\alpha(1-\alpha(\beta+\alpha(m+n))^{-1})L^{-1}),$$

$$\pi_1 = (1+\alpha)\lambda, \text{ and } \pi_2=(\beta+\alpha)(1-(1+m\alpha)\lambda)(\beta+\alpha(m+n))^{-1}-\alpha(\beta+\alpha)(\beta+\alpha(m+n))^{-1}L^{-1},$$

The proof of lemma 2 is given in the Appendix.

From (26), (29), and lemma 2 we have

$$A = (\alpha\,\pi_0 + (1+\alpha)\pi_1 + (\beta+\alpha)\pi_2)/2 \tag{30}$$

where π_0, π_1, and π_2 are as given in lemma 2.

Using the bounds on L, as given by (13), in (30) we have

$$d_1 + d_2\left(\frac{u(d_2)}{L_u^*} + \frac{1-u(d_2)}{L_\ell^*}\right) \triangleq A^* \le A \le A_u^* \triangleq d_1 + d_2\left(\frac{u(d_2)}{L_\ell^*} + \frac{1-u(d_2)}{L_u^*}\right) \tag{31}$$

where $d_1 = (\lambda(m\alpha^2+2\alpha+1)+(1-(1+m\alpha)\lambda)(\beta+\alpha(m+n))^{-1}((m+n)\alpha^2+\beta^2+2\alpha\beta))/2$

$d_2 = \alpha(\alpha-(\beta+\alpha(m+n))^{-1}(\alpha^2+(\beta+\alpha)^2)/2$

and where $u(d_2) = 1$ if $d_2 > 0$, and $u(d_2) = 0$ if $d_2 \le 0$.

In table 4 we give the values of the bounds A_ℓ^*, and A_u^* for representative values of the network parameters α and β. For each pair, (α,β), we used $(m,n)=(m^*,n^*)$, as given in table 1.

Next we consider the mean contention delay C. For $\lambda < \lambda_o(\alpha,\beta;m,n)$ we have that $C = V/\bar{S}$, where $\bar{S}$ is as given by (16). Thus,

$$C = (1-p_0)V/(\lambda L) \tag{32}$$

The bounds on V and L given by (25) and (13), respectively, can be used in (32) to obtain bounds on C. It is possible, however, to obtain bounds on V that are tighter than those given by (25). The method parallels the one dev oped in part (c) of section 3, and involves the computation of tighter bounds on V_k, for $1 \le k \le N$, where N is some finite natural number. Working towards this direction we express a theorem, parallel to theorem 2.

Theorem 6

Given α,β,m,n, let $\lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n))$; then, for every $N \ge 1$

$$\ell_1k^2 + \ell_2k + \ell_3 \le V_k^\ell \le V_k \le V_k^u \le u_1k^2 + u_2k + u_3\ ,\ 1 \le k \le N, \tag{33}$$

where $\{V_k^u\ ;\ 1\le k\le N\}$ is the unique solution to system (11) with

$$b_k = f_k(\{F_i\ ;\ i\ge 1\}) + \sum_{i=N+1}^{\infty} a_{k,i}(u_1k^2+u_2k+u_3),$$

and $\{V_k^{\ell}\ ;\ 1 \le k \le N\}$ is the unique solution to system (11) with

$$b_k = f_k(\{G_i\ ;\ i \ge 1\}) + \sum_{i=N+1}^{\infty} a_{k,i}(\ell_1 k^2 + \ell_2 k + \ell_3)\ ;$$

[illegible] $1 \le i \le N$, and $F_i = bk - c$, for $i > N$; $G_i = L_i^{\ell}$ for $1 \le i \le N$, [illegible]

[illegible] is parallel to the proof of theorem 2, and it is omitted.

[illegible] V_k^{ℓ}, for $1 \le k \le N$, and the quadratic bounds given by (22), [illegible]

$$V_\ell^{\star} \le V \le V_u^{\star} \tag{34}$$

$$V_u^{\star} = [illegible] + \frac{1}{1-p_0} \sum_{k=1}^{N} p_k(u_1 k^2 + u_2 k + u_3 - V_k^u) \tag{35}$$

$$V_\ell^{\star} = V_\ell + \frac{1}{1-p_0} \sum_{k=1}^{N} p_k(V_k^{\ell} - \ell_1 k^2 - \ell_2 k - \ell_3) \tag{36}$$

; where V_u, V_ℓ are as given in (25).

Using the bounds on V and L, as given by (34) and (13), respectively, in (32), yields the following bounds on C.

$$(1-p_0)V_\ell^{\star}/(\lambda L_u^{\star}) = C_\ell^{\star} \le C \le C_u^{\star} = (1-p_0)V_u^{\star}/(\lambda L_\ell^{\star}) \tag{37}$$

Finally, combining (37) with (31) yields the following bounds on the mean packet delay, for $\lambda \in [0, \lambda_0(\alpha, \beta; m, n))$.

$$D_\ell \le D \le D_u \tag{38}$$

where $D_u = A_u^{\star} + C_u^{\star}$, and $D_\ell = A_\ell^{\star} + C_\ell^{\star}$

We computed the bounds $\{V_k^u;\ 1\leq k\leq N\}$, and $\{V_k^\ell;\ 1\leq k\leq N\}$, required for the evaluation of V_u^*, and V_ℓ^*, by solving system (11), for N = 10, and with b_k, $1\leq k\leq 10$, as defined in theorem 6. We, subsequently, computed the bounds V_u^*, and V_ℓ^*, from (35), and (36), respectively. In all computations we used $m=m^*$, $n=n^*$ as given in table 1. In table 5, we give the bounds C_u^*, and C_ℓ^*, on the mean contention delay, as found from (37), for representative values of the network parameters α,β, and for λ in the range $[0, \bar{\lambda}(\alpha,\beta))$. Finally, in figures 6, and 7 we plot the bounds D_u, and D_ℓ on the mean packet delay, induced by the LANSA, with $(m,n)=(m^*,n^*)$, as found from (38), for representative values of the network parameters α,β. Note that the obtained bounds remain tight, (they coincide up to the fourth decimal point), even for λ close to $\bar{\lambda}(\alpha,\beta)$.

6. A GENERALIZED VERSION OF THE ALGORITHM AND SOME COMPARISONS

The LANSA is one of the simpler algorithms that can be designed to operate under limited channel sensing, yet it is inherently stable and attains high performance. This is achieved by, simply, parametrizing its operation on the parameters m and n, whose values are adjusted to the network characteristics for throughput maximization. The question that arises then is: If we parametrize the operation of the algorithm on a larger set of parameters, will this result in significant performance improvement? To answer this questions, we developed and analyzed a generalized version of the LANSA, named G-LANSA. The rules of the G-LANSA are the same with those of the LANSA, except for the following modifications.

Rule 1' -- Counter Initialization

$$CI_{k_1} = \begin{cases} 1 & , \text{ if } k_1 = k_0 \\ M & , \text{ if } k_1 \neq k_0\text{, and the channel was busy with a successful transmission} \\ \bar{M} & , \text{ if } k_1 \neq k_0\text{, and the channel was busy with a collision} \end{cases}$$

where M, and M' are integer valued random variables with distributions

$$P(M=i) = \begin{cases} \mu_i & \text{for } 1 \le i \le m, \\ 0 & \text{otherwise} \end{cases}, \qquad P(\bar{M}=i) = \begin{cases} \bar{\mu}_i & \text{for } 1 \le i \le \bar{m} \\ 0 & \text{otherwise} \end{cases}$$

and where $m \geq 1$ and $\bar{m} \geq 1$ are integer parameters.

<u>Rule 3' -- Counter updating (First alternative)</u>

a) If $CI_{k_i} > 1$, then

$$CI_{k_{i+1}} = \begin{cases} CI_{k_i} - 1 & \text{, if the channel was idle during } (k_i, k_{i+1}) \\ CI_{k_i} + m-1 & \text{, if a successful transmission occurred during } (k_i, k_{i+1}) \\ CI_{k_i} + \bar{m}+n-1 & \text{, if a collision occurred during } (k_i, k_{i+1}) \end{cases}$$

b) If $CI_{k_i} = 1$, and a collision occurred during (k_i, k_{i+1}), then

$$CI_{k_{i+1}} = Q$$

where Q is an integer valued random variable with distribution

$$P(Q=i) = \begin{cases} q_i & \text{for } 1 \le i \le \bar{m} + n \\ 0 & \text{otherwise} \end{cases}$$

If $CI_{k_i} = 1$, and a successful transmission occurred during (k_i, k_{i+1}), then the user has successfully transmitted his packet, and he becomes idle.

<u>Rule 3" - Counter updating (Second alternative)</u>

Same as rule 3' above except for the following modification. If $CI_{k_i} > 1$, and the channel was idle during (k_i, k_{i+1}), then

$$CI_{k_{i+1}} = CI_{k_i} - R$$

where

$$R = \begin{cases} 1 & \text{, if the last non-idle algorithm step before } (k_i,\, k_{i+1}) \text{ was successful} \\ 0 & \text{, if the last non-idle algorithm step before } (k_i,\, k_{i+1}) \text{ was unsuccessful, and } CI_{k_i} > 2 \\ \bar{R} & \text{, if the last non-idle algorithm step before } (k_i,\, k_{i+1}) \text{ was unsuccessful, and } CI_{k_i} = 2 \end{cases}$$

where $\bar{R} = 1$ with probability $\bar{p}$, and $\bar{R} = 0$ with probability $1 - \bar{p}$.

From rule 1' we see that the G-LANSA, in contrast to the LANSA, distinguishes a packet that arrived during a successful algorithm step from a packet that arrived during an unsuccessful algorithm step; more specifically, the former is placed in one of the first m cells of the stack according to the distribution $\{\mu_i, 1 \le i \le m\}$, whereas the latter is placed in one of the first $\bar{m}$ cells of the stack according to the distribution $\{\bar{\mu}_i,\ 1 \le i \le \bar{m}\}$. Since rule 1 of the LANSA is a special case of rule 1' with $m=\bar{m}$, and $\{\mu_i = \bar{\mu}_i = 1/m,\ 1 \le i \le m\}$, the performance of the G-LANSA will be at least as good as that of the LANSA. Note, however, that rule 1' requires that a user, who senses the channel upon his packet's arrival and finds it busy, should be able to distinguish between a successful transmission and a collision before the channel goes idle. In contrast, rule 1 is free of the above requirement, and, therefore, the LANSA is easier to implement, compared to the G-LANSA. Concerning rule 3', note that, when a collision occurs, the group of collided packets is split using the first $\bar{m}+n$ cells according to the distribution $\{q_i,\ 1 \le i \le \bar{m}+n\}$. Since the first $\bar{m}$ cells are also used to accommodate newly arrived packets, we see that the G-LANSA, in contrast to the LANSA, allows the "mixing" of new packets with collided packets in the first $\bar{m}$ cells of the stack. Also note that rule 3 of the LANSA is a special case of rule 3' with $m=\bar{m}$ and $\{q_i = 0,\ 1 \le i \le m;\ q_i = 1/n,\ m+1 \le i \le m+n\}$.

Finally, rule 3" presents an alternative counter updating scheme in the event of an idle algorithm step. Unlike rule 3', where a user always decrements his counter by one (i.e., R=1), rule 3" requires that users be more "cautious" and decrement their counters by either one or zero, depending on the past activity on the channel. The rationale for this cautiousness is that if the last non-idle algorithm step before the current idle step was unsuccessful, then the probability of a future collision is increased. Thus, to avoid this possible collision, packets in cell #2 (if any) are placed in cell #1 (i.e., are transmitted) with probability $\bar{p}$, or remain in cell #2 with probability $1-\bar{p}$, where $\bar{p}$ is a parameter to be optimized for throughput maximization. Rule 3" is similar in spirit to the "skip step" introduced by Massey, [10], to improve the performance of the original Capetanakis algorithm. For rule 3" to be implementable, users should maintain a "flag" that indicates whether the last non-idle algorithm step was successful or unsuccessful, since this determines the value of R used in the updating of the counter. We note that this is possible, even though users use only limited channel sensing.

Numerical Results and Throughput Comparisons

We analyzed the G-LANSA utilizing the methods used in the analysis of the LANSA. In table 6, we give the results for the lower bound $\bar{\lambda}$ on the maximum stable throughput attained by the G-LANSA that uses rule 3', for representative values of α, and for $\beta = 1$, $\beta = 0.5$ and $\beta = \alpha$. In the same table we include the best choices for the algorithmic parameters. In all cases the probabilities μ_i, $\bar{\mu}_i$, and q_i were chosen as follows:

$$\{\mu_i=1/m^*,\ 1\leq i\leq m^*\},\{\bar{\mu}_i=1/\bar{m}^*,\ 1\leq i\leq \bar{m}^*\},\{q_i = p^*/\bar{m}^*,\ 1\leq i\leq \bar{m}^*;\ q_i = (1-p^*)/n^*,\ \bar{m}^*+1\leq i\leq \bar{m}^*+n^*\}$$

, where the values of m^*, $\bar{m}^*$, n^*, and p^* are as given in table 6.

We should point out, however, that $\bar{\lambda}$ is not particularly sensitive to deviations from the given optimal parameter choices.

Comparison of tables 1 and 6 shows that, for β = 1, the G-LANSA with rule 3' coincides with the LANSA, since $m^*=\bar{m}^*$, and $p^*=0$; thus if the users do not have early collision detection capabilities, then, in the case of collision, "mixing" the collided users with the newcomers, i.e., using $p^*\neq 0$ does not improve the throughput performance. However, for smaller values of β, using $m^*\neq\bar{m}^*$, and $p^*\neq 0$ offers some performance improvement over the LANSA (the corresponding $\bar{\lambda}$'s differ in the third decimal point). The maximum performance improvement ($\bar{\lambda}$ increase in the second decimal point) over the LANSA is attained by the G-LANSA that uses rule 3", when $\beta = \alpha$. The results for $\bar{\lambda}$ and the optimal parameter values for this case are given in table 7.

In summary, the G-LANSA offers a slight performance improvement over the LANSA, at the expense of increased operational complexity. Thus, unless β is very close to α, the LANSA is practically sufficient.

In figure 8 we plot the lower bound $\bar{\lambda}$ on the maximum stable throughput of the LANSA and the G-LANSA as a function of α, for β = 1, along with the maximum stable throughput attained by the controlled NP-CSMA algorithms of [5], and the Window-CSMA algorithms of [12]. Figure 9 presents the corresponding results for $\beta = \alpha$. We should point out that in the controlled NP-CSMA algorithms it is assumed that the network users are aware of the number of backlogged packets currently in the system. However, this information is not available to the users, and must be estimated.

The Window-CSMA algorithms are the extension of Gallager's algorithm [9], which is the most efficient Tree algorithm known to date, to the carrier sensing environment considered here. The Window-CSMA algorithms use continuous channel sensing; that is, users are required to sense the channel constantly even if they have nothing to transmit.

As it can be seen from figures 8, and 9, the proposed algorithms out perform the controlled NP-CSMA algorithms for every value of the propagation delay α. The maximum stable throughput attained by the Window-CSMA algorithms is slightly higher

than the lower bound $\bar{\lambda}$ for larger values of α, but it becomes slightly lower than $\bar{\lambda}$ as α decreases. Thus, generally speaking, the proposed algorithms attain throughputs as high as the Window-CSMA algorithms, despite the fact that the former use limited channel sensing, and are much easier to implement than the latter.

So far we have compared the throughput performance of the LANSA and the G-LANSA to the performance of two other heuristic algorithms. The question that arises then is: what is the maximum stable throughput that can be achieved by the optimal algorithm, (i.e., the algorithm that attains the greatest maximum stable throughput) in the class of algorithms that operate under the user - channel model described in section 2. Given a user - channel model, the maximum stable throughput of such an optimal algorithm is termed the capacity of the user and channel model. Considering the class of all realizable algorithms that do not use short packets to reserve the channel, Molle in [11] and Humblet in [24] have derived upper bounds on the capacity of the user - channel model considered here. In [25], we have used the bounding techniques developed in [26] to derive tighter bounds than those of [11], and [24]. These upper bounds on the capacity are included in figures 8, and 9, for $\beta = 1$, $\beta = \alpha$, respectively.

7. CONCLUSIONS

In this paper we presented "limited sensing" random access algorithms for carrier sense multiple access channels. The proposed algorithms are representative members of a new class of stable algorithms with "limited sensing" and "free access" characteristics. "Limited sensing" algorithms require that users sense the channel only while they have a packet to transmit, and, therefore, they have practical advantages over algorithms that require continuous channel sensing, such as the algorithms in [12]. The term "free access" is used to denote the fact that a user may transmit a packet immediately after its generation, provided that he senses the channel idle; this latter feature simplifies the implementation of the algorithm even more.

We demonstrated that the proposed algorithms combine simplicity and limited channel sensing with inherently stable operation and high performance. More specifically, we derived lower bounds on the maximum stable throughput induced by both the basic algorithm, named LANSA, and its generalization, named G-LANSA. On the basis of the derived bounds, we concluded that the proposed algorithms out perform the optimally controlled version of the traditional non-persistent CSMA algorithms [5], and have similar throughput characteristics with the most efficient Tree-type algorithms [12], despite the fact that the latter use continuous channel sensing and are more complex to implement.

We introduced bounding techniques that can yield arbitrarily tight upper and lower bounds on the induced mean packet delay. We used these techniques to evaluate the mean packet delay induced by the LANSA. The delay analysis exploited the existence of points in time, where the stochastic processes associated with the random access system probabilistically restart themselves. Using the theory of regenerative processes we showed that the LANSA is stable if it induces persistent regeneration points, with finite mean recurrence time, (i.e., $L < \infty$). Moreover, it was shown that under the same condition the various "averages" concerning the packet delay (limiting average, limiting expected average, and expectation with respect to limiting distribution) exist and coincide. Many of the random access algorithms encountered in the literature have regenerative properties. Thus, the direction taken in this paper may be used in the stability analysis, and in the evaluation of the mean packet delay of several other schemes. In this study we dealt only with the mean packet delay. However, to fully characterize the delay performance offered by the network, knowledge of the packet delay probability distribution is needed. At this point, the analytical evaluation of the delay distribution induced by the algorithms presented here seems extremely hard. Note, however, that the bounding techniques used in this paper can be extended to yield arbitrarily tight upper and lower bounds on the higher moments of the delay [28]. We should

point out here that the LANSA and its generalization have last come - first served characteristics (in a generalized sense); that is, new arrivals enter the system upon arrival, and they are usually accommodated before packets already in the system are. This characteristic causes relatively large variances in the induced delay, but it also favors "impatient" users. The favoritism to impatient users is advantageous in networks in which packets must either be transmitted within a short time limit or be lost [20]. However, to fully explore the last come - first served characteristics of the limited sensing algorithms, additional research is needed. The behavior of such algorithms must be studied when specific upper limits on delays are imposed, and when packets in different stages of algorithmic progress depart the system.

Finally, we note that the algorithms presented in this paper have been modified to operate asynchronously in [27]. The asynchronous (unslotted) algorithms simplify the operation of the network, since there is no need for the users to maintain a global time base.

In closing, we mention that, very recently, Humblet [29], and Georgiadis et al. [30] have independently shown that Tree algorithms can be modified to operate under limited channel sensing. Unlike the algorithms proposed here, their algorithms have "limited sensing" and "blocked access" characteristics. This is another class of inherently stable algorithms, that combines high performance with simplicity, and it should be further explored in the carrier-sense environment.

APPENDIX

Proof of Theorem 2

Part (i)

To prove that system (6) has a non-negative bounded solution, $\{y_k\}_{k\geq 0}$, we construct a sequence $\{y_k^{(0)}\}_{k\geq 0}$, that serves as an upper bound to this solution. With b and c appropriate real constants, we define

$$y_0^{(0)} \triangleq \alpha \ ; \ y_k^{(0)} \triangleq bk-c \ , \ k \geq 1 \ . \tag{A.1}$$

We also define the sequences $\{y_k^{(n)}\}_{k\geq 0}$, $n\geq 1$, as follows

$$y_0^{(0)} \triangleq \alpha \ ; \ y_k^{(n)} \triangleq \sum_{i=1}^{\infty} a_{k,i} \, y_i^{(n-1)} + g_k \ , \ k \geq 1 \tag{A.2}$$

where the coefficients $a_{k,i}$ and g_k are as defined in the theorem.

In a straightforward manner we obtain

$$y_k^{(1)} = y_k^{(0)} - d_k \ , \ k \geq 0 \tag{A.3}$$

where $d_0 = 0$, and d_k, $k \geq 1$, are as given in (A.5) and (A.6). Since the coefficients $a_{i,k}$ are non-negative, we deduce from (A.1), (A.2) and (A.3) that for every fixed $k \geq 0$, the sequence $\{y_k^{(n)}, \ n\geq 0\}$ will be non-negative and non-increasing, if we choose b and c such that the following inequalities are satisfied:

$$bk-c \geq 0, \text{ for every } k \geq 1, \ (b>0). \tag{A.4}$$

$$d_1 \triangleq (1-(+m\alpha)\lambda) \ b - (1-m(1-s_0))c - (1+\alpha(1+ms_0)) \geq 0 \tag{A.5}$$

$$d_k \triangleq -\lambda(\beta+(m+n)\alpha)b + G_k \ c-(\beta+\alpha(1+mr_0+nq_{k,0})) \geq 0, \text{ for every } k \geq 1 \tag{A.6}$$

where $G_k \triangleq (1-r_0)m + (1-q_{k,0})n-1$

Thus, under conditions (A.4), (A.5), and (A.6), the following limits exist:

$$\lim_{n\to\infty} y_k^{(n)} = y_k \text{ , } k \geq 0 \tag{A.7}$$

where $y_0 = \alpha$, and $y_k \leq bk-c$, $k \geq 1$.

The numbers y_k obtained in this manner solve system (6). Indeed, let us pass to the limit as $n\to\infty$ in equation (A.2). In the right hand side a term-by-term passage to the limit is admissible, for the series at the right converges uniformly as regards n, since it is bounded from above by the series with constant terms $y_k^{(0)} = bk-c$. Thus, on effecting this passage we find that

$$y_k = \sum_{i=1}^{\infty} a_{k,i}\, y_i + g_k \text{ , } k \geq 1$$

; i.e., $\{y_k, k\geq 0\}$ is indeed a solution to system (6).

Next we investigate the conditions under which inequalities (A.4), (A.5), and (A.6) are satisfied. First we prove that $G_k > 0$, for every $k \geq 2$, $m \geq 1$, $n \geq 2$, $\alpha \geq 0$, $\beta \geq 0$, and $\lambda \geq 0$. We have

$$G_k \geq (1-q_{k,0})n-1 \geq (1-q_{2,0})n-1 \geq (1-(1-n^{-1})^2)n-1 = 1-n^{-1} > 0 \tag{A.8}$$

Also, from the well known inequality $\exp(-x) \geq 1-x$, we have that for $\lambda \geq 0$

$$1-m(1-s_0) \geq 1-(1+m\alpha)\lambda \tag{A.9}$$

From (A.6) and (A.8) we have that c must be positive. If $1-(1+m\alpha)\lambda \leq 0$ and $1-m(1-s_0) \geq 0$, then condition (A.5) cannot be true, since $b > 0$ and $c > 0$. If $(1-(1+m\alpha)\lambda \leq 0$ and $1-m(1-s_0) < 0$, then to satisfy condition (A.5) we must choose $c > b$, which contradicts condition (A.4). Thus,

$$\lambda < (1+m\alpha)^{-1} \tag{A.10}$$

Next we choose c and b such that (A.5) is met with equality; that is,

$$c = ((1-(1+m\alpha)\lambda)b - (1+\alpha(1+ms_0))(1-m(1-s_0))^{-1} \tag{A.11}$$

The above choice also guarantees that condition (A.4) is met, since, according to (A.9) and (A.10), we have

$$0 < (1-(1+m\alpha)\lambda)(1-m(1-s_0))^{-1} < 1,$$

and therefore b > c.

Substituting (A.11) into (A.6) yields,

$$G_k(u_k\, b-v_k) \geq 0 \ , \ k \geq 2 \tag{A.12}$$

where

$$v_k=v_k(\lambda)=(1+\alpha(1+ms_0))(1-m(1-s_0))^{-1}+(\beta+\alpha(1+mr_0+nq_{k,0}))((1-r_0)m+(1-q_{k,0})n-1)^{-1}$$

$$u_k=u_k(\lambda)=(1-(1+m\alpha)\lambda)(1-m(1-s_0))^{-1}-\lambda(\beta+(m+n)\alpha)((1-r_0)m+(1-q_{k,0})n-1)^{-1}$$

Since $G_k > 0$, since u_k is a monotone increasing function of k, and since $v_k > 0$, $k \geq 2$, we have that condition (A.12) is met, if $u_2 > 0$, and if $b \geq \max\{v_k/u_k\}_{k\geq 2}$. Since the ratio v_k/u_k is a monotone decreasing function of k, we choose $b = v_2/u_2$.

Next we show that, given α,β,m, and n, the condition $u_2(\lambda) > 0$ is met if $\lambda < \lambda_o(\alpha,\beta;m,n)$, where $\lambda_o(\alpha,\beta;m,n)$ is the unique root of equation (7) over the interval $[0,(1+m\alpha)^{-1})$. First we write $u_2(\lambda) = F(\lambda)/G_2(\lambda)$, where $F(\lambda)\triangleq(1-(1+m\alpha)\lambda)\cdot(1-m(1-s_0))^{-1}G_2(\lambda)-\lambda(\beta+(m+n)\alpha)$. From (A.10) we have that $G_2(\lambda) > 0$, for every $\lambda \geq 0$. Thus, we examine only the function $F(\lambda)$. For this function it can be easily proved that $d^2F(\lambda)/d\lambda^2 < 0$, for every $\lambda\varepsilon[0,(1+m\alpha)^{-1}]$; that is, the $F(\lambda)$ is a concave function over the domain of interest. Now, since $F(0) > 0$ and $F((1+m\alpha)^{-1}) < 0$, we clearly have that the equation $F(\lambda) = 0$, or equivalently equation (7) in the theorem, has a unique root, $\lambda_o(\alpha,\beta;m,n)$, in the interval $[0,(1+m\alpha)^{-1})$. Furthermore, for all $\lambda\varepsilon[0,\lambda_o(\alpha,\beta;m,n))$ we have that $F(\lambda) > 0$, or equivalently

$$u_2(\lambda) = (1-(1+m\alpha)\lambda)(1-m(1-s_0))^{-1}-\lambda(\beta+(m+n)\alpha)G_2^{-1}(\lambda) > 0 \tag{A.13}$$

The construction of the lower bound, $b'k-c'$, on y_k is parallel to that of the upper bound and is omitted. We only note that if we let

$$c' = ((1-(1+m\alpha)\lambda)b'-(1+\alpha(1+ms_0))(1-m(1-s_0))^{-1}$$

then, instead of condition (A.12), we now have the condition $G_k(u_k b'-v_k) \leq 0$, $k \geq 2$, which is satisfied if we choose $b' = \min\{v_k/u_k\}_{k\geq 2} = v_\infty/u_\infty$.

Part (ii)

Here we show that the mean session lengths, $\{L_k\}_{k\geq 0}$, induced by the algorithm coincide with the solution $\{y_k\}_{k\geq 0}$ of part (i). The proof is intimately related to the uniqueness of a solution to an infinite dimensional linear system, such as system (6). We should point out that the question of the uniqueness of the solution depends upon what conditions are imposed on the solution. The following lemma indicates a class of sequences in which the solution of system (6) is unique.

Lemma A:

In the class of non-negative sequences $\{z_k\}_{k\geq 0}$ for which

$$\sup_{k\geq 1}\left(\frac{z_k}{k^2}\right) < \infty \tag{A.14}$$

system (6) has a unique solution.

Proof Suppose that system (6) has two non-negative solutions satisfying (A.14), and let the sequence $\{w_k\}_{k\geq 0}$ denote their difference. Then, the sequence $\{w_k\}_{k\geq 0}$ satisfies condition (A.14), and solves the following system:

$$w_0 = 0 \; ; \; w_k = \sum_{i=1}^{\infty} a_{k,i}\, w_i \; , \; k \geq 1$$

Next define $\bar{w}_0 \triangleq 0$, and $\bar{w}_k \triangleq w_k/(u_1k^2+u_2k+u_3)$ for $k \geq 1$, where u_1, u_2, u_3 are real constants with $u_1 > 0$. The sequence $\{\bar{w}_k\}_{k\geq 0}$ solves the following system:

$$\bar{w}_0 = 0 \; ; \; \bar{w}_k = \sum_{i=1}^{\infty} \bar{a}_{k,i}\, \bar{w}_i \; , \; k \geq 1$$

where $\bar{a}_{k,i} = a_{k,i}\,(u_1i^2+u_2+u_3)/(u_1k^2+u_2k+u_3)$, $i \geq 1$, $k \geq 1$.

We shall prove that there exist coefficients u_1, u_2, u_3, ($u_1 > 0$), such that

$$\sup_{k \geq 1} \sum_{i=1}^{\infty} |a_{ki}| < 1 \tag{A.15}$$

Assume that u_1, u_2, u_3 are chosen such that

$$Q(k) = u_1 k^2 + u_2 k + u_3 > 0 \text{ , for every } k \geq 1 \tag{A.16}$$

The following conditions are, clearly, sufficient for the validity of (A.16).

$$u_1 > 0,\ Q(1) = u_1 + u_2 + u_3 > 0, \text{ and } Q'(1) = 2\, u_1 + u_2 \geq 0 \tag{A.17}$$

Under condition (A.16), and after straightforward calculations we have

$$\sum_{i=1}^{\infty} |\bar{a}_{ki}| = \sum_{i=1}^{\infty} \bar{a}_{ki} = \frac{d_{k1}u_1 + d_{k2}u_2 + d_{k3}u_3}{u_1 k^2 + u_2 k + u_3} \triangleq \xi_k,\ k \geq 1$$

; where $d_{k1} = k^2 - c_{k1}$, $d_{k2} = k - c_{k2}$, $d_{k3} = 1 - c_{k3}$, and where

$$c_{11} = 1 + (1+m\alpha)\lambda(1+(\alpha+m^{-1})\lambda),\ c_{12} = 1-(1+m\alpha)\lambda,\ c_{13} = 1-m(1-s_0)$$

$$c_{k1} = (1-n^{-1})k^2 - (2\alpha\lambda+1-n^{-1})k - n\alpha\lambda(1+\alpha\lambda) - (\beta+m\alpha)\lambda(1+(\beta m^{-1}+\alpha)\lambda) \text{ , } k \geq 2$$

$$c_{k2} = -\lambda(\beta+\alpha(m+n)),\ c_{k3} = 1-m(1-r_0)-n(1-q_{k,0}) \text{ , } k \geq 2.$$

Next we prove that $\xi_k < 1$, or equivalently that

$$c_{k1}u_1 + c_{k2}u_2 + c_{k3}u_3 > 0 \text{ , } k \geq 1 \tag{A.18}$$

Assuming that $u_1 > 0$, we rewrite conditions (A.17) and (A.18) as follows

$$\phi + \psi + 1 > 0,\ \phi + 2 \geq 0 \text{ , } c_{k2}\,\phi + c_{k3}\psi + c_{k1} > 0,\ k \geq 1 \tag{A.19}$$

where $\phi = u_2/u_1$, and $\psi = u_3/u_1$.

Given α,β,m,n, then, for every $\lambda \varepsilon [0, \lambda_o(\alpha,\beta;m,n))$, we have that

$$0 < c_{12} < c_{13} \text{ ; } c_{k2} < 0,\ c_{k3} < 0,\ k \geq 2 \tag{A.20}$$

; furthermore, it follows from (A.14) that, for every $k \geq 2$,

$$0 \le \frac{c_{k2}}{c_{k3}} < \frac{c_{12}}{c_{13}} < 1 \tag{A.21}$$

From (A.20), and (A.21) we deduce that the inequalities given in (A.19) are satisfied for any ϕ and ψ, such that

$$\phi > \max\left(-2,\left(\frac{c_{11}}{c_{13}} - 1\right)\cdot\left(1 - \frac{c_{12}}{c_{13}}\right)^{-1}, \left(\frac{c_{k1}}{c_{k3}} - \frac{c_{11}}{c_{13}}\right)\cdot\left(\frac{c_{12}}{c_{13}} - \frac{c_{k2}}{c_{k3}}\right)\right), \ k \ge 2.$$

$$-\frac{c_{12}}{c_{13}}\phi - \frac{c_{11}}{c_{13}} < \psi < -\frac{c_{k2}}{c_{k3}}\phi - \frac{c_{k1}}{c_{k3}}, \ k \ge 2$$

; furthermore, inequalities (A.20) and (A.21), guarantee that such ϕ and ψ always exist.

Thus, we have proved that there exist u_1, u_2, u_3 such that

$$\xi_k = \sum_{i=1}^{\infty} |\bar{a}_{ki}| < 1 \ , \text{ for every } k \ge 1 \tag{A.22}$$

Also, from the functional form of d_{k1}, d_{k2}, and d_{k3} we deduce that

$$\lim_{k\to\infty} \xi_k = \frac{1}{n} < 1 \tag{A.23}$$

Given $n \ge 2$, we choose an $\varepsilon > 0$, such that $\varepsilon < \min(n^{-1}, 1-n^{-1})$. Then it follows from (A.23) that there exists $k_0 \ge 1$, such that

$$|\xi_k - \frac{1}{n}| < \varepsilon, \text{ for every } k > k_0 \tag{A.24}$$

If we let

$$M = \max_{1\le k\le k_0} (\xi_k) \tag{A.25}$$

then, from (A.22), we have that $M < 1$. Thus, it follows from (A.24), and (A.25) that

$$h \overset{\Delta}{=} \sup_{k \geq 1} \{\xi_k\} \leq \max(\frac{1}{n} + \varepsilon,\ M) < 1$$

Now, let ρ denote the supremum of $|\bar{w}_k|$, $k \geq 1$. From (A.14), we have that $\rho < \infty$, and that

$$|\bar{w}_k| \leq \sum_{i=1}^{\infty} |\bar{a}_{ik}|\ \rho \leq h\rho, \text{ for every } k \geq 1.$$

Thus, $\rho \leq h\rho$, and since $h < 1$, we have $\rho = 0$. Thus, $|\bar{w}_k| = 0$ for every $k \geq 1$; i.e., $|w_k| = 0$ for every $k \geq 1$, and the proof of the lemma is complete.

The solution X of part (i) clearly satisfies condition (A.14), thus, from lemma A, we have that X is the unique non-negative solution to system (6) satisfying condition (A.14). Next we use arguments parallel to those used in Theorem 6 of [13], to show that $y_k = L_k$, for every $k \geq 0$. Let us consider the random variable $\ell_\tau = \min(\ell,\tau)$, where ℓ is the session length, and τ is a real number, $\tau > 0$. Also, let $L_k(\tau) = E(\ell_\tau | K=k)$, $k \geq 0$, where K is the multiplicity of the session. It can be easily seen that

$$0 \leq L_k(\tau_1) \leq L_k(\tau_2), \text{ for any } \tau_1,\tau_2, \text{ such that } 0 < \tau_1 \leq \tau_2 \tag{A.26}$$

$$\lim_{\tau\to\infty} L_k(\tau) = L_k,\ k \geq 0 \tag{A.27}$$

Since $L_k(\tau) \leq \tau$, from (6) and (A.26) we deduce that

$$0 \leq L_0(\tau) \leq \alpha;\ 0 \leq L_k(\tau) \leq \sum_{i=1}^{\infty} a_{k,i}\ L_i(\tau) + g_k\ ,\ k \geq 1$$

As was done in proving part (i) of the theorem, it can be shown that the solution $\{y_k\}_{k \geq 0}$ majorizes $L_k(\tau)$, $k \geq 0$, for any fixed τ; that is, $L_k(\tau) \leq y_k$, $k \geq 0$. Thus, from (A.26), (A.27) we deduce that $L_k \leq y_k$, $k \geq 0$, and, therefore, the mean session length sequence $\{L_k\}_{k \geq 0}$ satisfies condition (A.14). Using the result in Lemma A, we have $L_k = y_k$, for every $k \geq 0$.

Proof of Lemma 1

Given $N \geq 1$, and $b_k \geq 0$, $1 \leq k \leq N$, we define the sequences $\{x_k^{(n)}\}_{1 \leq k \leq N}$, $n \geq 0$, such that

$$x_k^{(0)} = \gamma k-\delta \ , \ x_k^{(n+1)} \triangleq \sum_{i=1}^{N} a_{k,i} \ x_i^{(n)} + b_k \ , \ 1 \leq k \leq N, \tag{A.28}$$

where γ, δ are real constants.

As was done in the proof of theorem 2, it can be shown that if $\lambda < \lambda_o(\alpha,\beta;m,n)$ then there exist γ and δ, $\gamma > \delta > 0$, such that $x_k^{(1)} \leq x_k^{(0)} = \gamma k-\delta$, for every k, $1 \leq k \leq N$; (see [25] for details). Thus, from (A.28) we have that, for every $n \geq 0$, $0 \leq x_k^{(n+1)} \leq x_k^{(n)}$, $1 \leq k \leq N$. Thus, the limit $x_k = \ell im \ x_k^{(n)}$, as $n\to\infty$, exists and is non-negative. Finally, taking limits, as $n\to\infty$, in both sides of (A.28) proves that the sequence $\{x_k\}_{1 \leq k \leq N+1}$ solves system (11). To complete the proof of the lemma we use the following standard result in the theory of finite linear systems with non-negative coefficients (see, for example [11, Thm. 2.1]): if system (11) has a non-negative solution for every $b_k \geq 0$, $1 \leq k \leq N$, then the matrix $(I_N-A_N)^{-1}$ exists, and has non-negative elements. Thus, given $b_k \geq 0$, $1 \leq k \leq N$, system (11) has a unique non-negative solution $\underline{x} = (I_N-A_N)^{-1}\underline{b}$, where $\underline{x} = (x_1,\ldots,x_N)^t$, and $\underline{b} = (b_1,\ldots,b_N)^t$.

Proof of Theorem 3

Let $\{x_k\}_{1 \leq k \leq N}$ and $\{x_k'\}_{1 \leq k \leq N}$ be the solution to system (11) that corresponds to the non-negative sequence $\{b_k\}_{1 \leq k \leq N}$ and $\{b_k'\}_{1 \leq k \leq N}$, respectively. If $b_k \leq b_k'$, $1 \leq k \leq N$, then from lemma 1 we have that $x_k \leq x_k'$, $1 \leq k \leq N$. This monotone increasing property of the solutions to system (11) with respect to the non-negative forcing terms proves the theorem. For example, to prove that $L_k \leq L_k^u$, $1 \leq k \leq N$, we argue as follows. From proposition 2, we have that $\{L_k\}_{1 \leq k \leq N}$ solves system (11) with forcing terms

$$b_k' = \sum_{i=N+1}^{\infty} a_{k,i} \ L_i + g_k \ , \ 1 \leq k \leq N$$

If $\lambda < \lambda_o(\alpha,\beta;m,n)$, then from theorem 2 we have that $b_k' \geq b_k$, where the coefficients

b_k, $1 \leq k \leq N$, are as defined in theorem 3. Thus, from the monotonicity of the solutions, we have that $L_k \leq L_k^u$, $1 \leq k \leq N$. The lower bounds $\{L_k^{\ell}\}_{1\leq k\leq N}$ are established in a similar manner.

Proof of Theorem 5

Part (i)

As it can be seen from the proof of theorem 2, to prove that system (21) has a solution $\{z_k\}_{k\geq 0}$, such that $0 \leq z_k \leq u_1k^2+u_2k+u_3$, $k \geq 1$, it suffices to show that there exist u_1, u_2, u_3, such that $u_1k^2+u_2k+u_3 \geq 0$ and

$$\sum_{i=1}^{\infty} a_{k,i}(u_1i^2+u_2i+u_3) + f_k(\{L_j\}_{j\geq 0}) \leq u_1k^2+u_2k+u_3, \text{ for every } k \geq 1 \tag{A.29}$$

From theorem 2 we have that if $\lambda < \lambda_o(\alpha,\beta;m,n)$, then $0 \leq L_k \leq y_k^{(0)}$, where $y^{(0)} = \alpha$ and $y_k^{(0)} = bk-c$, $k \geq 1$. Thus, inequality (A.29) will be true if

$$0 \leq \sum_{i=1}^{\infty} a_{k,i}(u_1i^2+u_2i+u_3) + h_k \leq u_1k^2+u_2k+u_3 \text{ , for every } k \geq 1, \tag{A.30}$$

where $h_k = f_k(\{y_j^{(0)}\}_{j\geq 0})$. After straightforward manipulations, inequality (A.30) becomes

$$c_{k1}u_1 + c_{k2}u_2 + c_{k3}u_3 \geq h_k \text{ , } k \geq 1, \tag{A.31}$$

where c_{k1}, c_{k2}, c_{k3}, $k \geq 1$, are as defined in the proof of theorem 2, where

$$h_1 = 1 + \alpha + \frac{1}{2}\lambda(m-1)(\lambda(\alpha+\frac{1}{m})b - (1-s_0)c + \alpha\, s_0)$$

$$h_k = \frac{1}{2}(1-\frac{1}{n})bk^2 + \frac{1}{2}(n-1)(b(\alpha\lambda-\frac{1}{n})-c)k + (b\lambda(\beta+m\alpha)-mc+mr_0(\alpha+c))(k+\frac{1}{2}\beta\lambda(1-\frac{1}{m}))$$
$$+ (\beta+\alpha)k + \frac{n}{2}(\alpha+c)k\, q_{k,0} \text{ , } k \geq 1$$

and where b, c are as defined in theorem 2.

If $\lambda < \lambda_o(\alpha,\beta;m,n)$, then from (A.18) we have that there exist real constants u_1', u_2', u_3', such that $c_{k1}u_1'+c_{k2}u_2'+c_{k3}u_3' > 0$ for every $k \geq 1$; furthermore

$u_1'k^2+u_2'k+u_3'>0$, $k \geq 1$. If we define $\Theta \overset{\Delta}{=} \sup\{h_k/(c_{k_1}u_1'+c_{k_2}u_2'+c_{k_3}u_3'),k\geq 1\}$ then $0<\Theta<\infty$, since $h_k > 0$ for every $k \geq 1$, and since

$$\lim_{k\to\infty} (h_k/c_{k1}u_1'+c_{k2}u_2'+c_{k3}u_3')) < \infty$$

Thus, if we choose $u_1 = \Theta\, u_1'$, $u_2 = \Theta\, u_2'$, and $u_3 = \Theta\, u_3'$, then inequality (A.31) is satisfied, furthermore $u_1k^2+u_2k+u_3 > 0$, $k \geq 1$. Thus, system (21) was a non-negative solution $\{z_k\}_{k\geq 0}$, such that $z_0 = 0$, and $0 \leq z_k \leq u_1k^2+u_2+u_3$, $k \geq 1$.

The lower bounds $\{\ell_1k^2+\ell_2k+\ell_3\}_{k\geq 1}$ on the solution can be established in a similar manner.

Part (ii)

The proof is parallel to the proof of part (ii) of theorem 2, and is omitted.

Proof of Lemma 2

The process $\{Z(t),\ t\geq 0\}$ probabilistically restarts itself at the algorithm renewal instants, R_n, $n \geq 1$. Thus, it is regenerative with respect to the sequence $\{R_n\}_{n\geq 1}$, with regeneration cycle coinciding with the LANSA session. The following is a standard result relating to the regenerative process $\{Z(t),\ t\geq 0\}$

$$\pi_j \overset{\Delta}{=} \lim_{t\to\infty} P(Z(t)=j) = \frac{E(\text{amount of time in state } j \text{ during one session})}{E(\text{time of one session})} \tag{A.32}$$

where j=0,1, or 2. Furthermore, if we associate the process $\{Z(t),\ t\geq 0\}$ with the Poisson arrival process, then it can be shown (see, [18, Thm. 3]) that

$$\lim_{n\to\infty} P(Z(a_n)=j) = \lim_{t\to\infty} P(Z(t)=j) = \pi_j\ ,\ j=0,1,2.$$

We now proceed with the evaluation of the limiting probabilities, π_j, j=0,1,2. Let I, S, and U, denote the expected number of idle, successful, and unsuccessful algorithm steps over the course of a session. Then, in view of (A.32), and for $\lambda < \lambda_o(\alpha,\beta;m,n)$, we have

$$\pi_0 = \alpha\ I/L;\ \pi_1 = (1+\alpha)\ S/L;\ \pi_2 = (\beta+\alpha)U/L \tag{A.33}$$

where L is the mean session length.

Consider next an arbitrary session induced by the LANSA, and recall that the session starts with the marker set to cell #2, and that it ends when the marker drops to cell #1 for the first time. Since the marker's position is incremented by $m-1$, or $m+n-1$ after each successful, or unsuccessful algorithm step, respectively, and it is decremented by one after each idle algorithm step, we have that

$$I = 1 + (m-1)\ S + (m+n-1)\ U \tag{A.34}$$

Also, since an idle, successful, or unsuccessful step lasts for α, $1+\alpha$, or $\beta+\alpha$ units of time, respectively, we have that

$$L = \alpha\ I + (1+\alpha)\ S + (\beta+\alpha)\ U \tag{A.35}$$

Then, using equations (A.34), (A.35), and the fact that $S = \lambda L$, in (A.33), we find the expressions of the limiting probabilities given in the lemma.

REFERENCES

[1] L. Kleinrock and F.A. Tobagi, "Packet switching in radio channels: Part I - Carrier sense multiple access modes and their throughput-delay characteristics," IEEE Trans. Comm., vol. COM-23, pp. 1400-1416, Dec. 1975.

[2] F.A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part IV - Stability considerations and dynamic control in carrier sense multiple access," IEEE Trans. Comm., vol. COM-25, pp. 1103-1119, Oct. 1977.

[3] F.A. Tobagi and V.B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," Comput. Nets., vol. 4, pp. 243-249, Oct.Nov. 1980.

[4] R.M. Metcalfe and D.R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," Commun. Ass. Comput. Mach., vol. 19, pp. 395-903, July 1976.

[5] J.S. Meditch, and C.T.A. Lea, "Stability and optimization of the CSMA and CSMA/CD channels," IEEE Trans. Comm., vol. COM-31, no. 6, pp. 763-773, June 1983.

[6] J.F. Hayes, "An adaptive technique for local distribution," IEEE Trans. Comm., vol. COM-26, pp. 1178-1186, Aug. 1978.

[7] J.I. Capetanakis, "Tree algorithms for packet broadcast channels," IEEE Trans. Inform. Theory, vol. IT-25, pp. 505-515, Sept. 1979.

[8] B.S. Tsybakov and V.A. Mikhailov, "Slotted multiaccess packet broadcasting feedback channel," Problemy Peredachi Informatsii, vol. 14, no. 4, Oct./Dec. 1978.

[9] R.G. Gallager, "Conflict resolution in random access broadcast networks," presented at AFOSR Workshop on Commun. Theory, Provincetown, MA, 1978.

[10] J.L. Massey, "Collision resolution algorithms and random access communications," Multi-User Communications (CISM Courses and Lectures Series), G. Longo, Ed., New York: Springer-Verlag, 1981.

[11] M.L. Molle, "Unifications and extensions of the multiple access communications problem," CSD Report No. 810730 (UCLA-ENG-8118), Computer Science Department, University of California, Lost Angeles (July 1981), Ph.D. Dissertation.

[12] D. Towsley and G. Venkatesh, "Window random access protocols for local computer networks," IEEE Trans. Comp., vol. C-31, pp. 715-722, Aug. 1982.

[13] B.S. Tsybakov and N.D. Vvedenskaya, "Stack algorithm for random multiple access," Problemy Peredachi Informatsii, vol. 15, no. 3, pp. 80-94, Jul./Sept. 1980.

[14] L. Merakos, "Limited sensing random multiple access using binary feedback," in Proc. Conf. Inform. Sci. Syst., Baltimore, MD, March 1983, pp. 557-563.

[15] P. Mathys, and P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free or blocked channel access," in Abst. Papers, 1983 IEEE Inform. Theory Symp., St. Jovite, Quebec, Canada, Sept. 1983.

[16] N.D. Vvedenskaya, and B.S. Tsybakov, "Random multiple access of packets to a channel with errors," Problemy Peredachi Informatsii, vol. 19, no. 2, pp. 52-68, Apr.-Jun. 1983.

[17] S.M. Ross, Stochastic Processes, New York: Wiley, 1983.

[18] S. Stidham, Jr., "Regenerative processes in the theory of queues, with applications to the alternating-priority queue," Adv. Appl. Probl., vol. 4, pp. 542-577, 1972.

[19] F.J. Beutler, "Mean sojourn times in Markov queueing networks: Little's formula revisited," IEEE Trans. Inform. Theory, vol. IT-29, no. 2, pp. 233-241, March 1983.

[20] J.F. Kurose, and M. Schwartz, "A family of window protocols for time constrained applications in CSMA Networks," Proc. IEEE Inforcom '83, San Diego, CA, April 1983, pp. 405-413.

[21] J.F. Shoch, Y.K. Dalal, and D.D. Redell, "Evolution of the Ethernet local computer network," Computer, Aug. 1982, pp. 1-27.

[22] H. Kobayashi, T. Hirose and H. Haruyama, "Broadband network CSMA/CD compatible with baseband systems," Proc. IEEE Globecom' 84, Atlanta, Georgia, Nov. 1984, pp. 1058-1063.

[23] Local Network Standards, Draft C, IEEE Project 802, May 1982.

[24] P.A. Humblet, "Bounds on the utilization of ALOHA-like multiple-access broadcast channels," LIDS-P-1000, MIT Laboratory for Information and Decision Systems, June 1980.

[25] L. Merakos, "Stack algorithms for local area networks using CSMA channels," Technical Report EECS-TR-84-6, EECS Dept., University of Connecticut, June 1984.

[26] V.A. Mikhailov and B.S. Tsybakov, "Upper bound for the capacity of a random multiple access system," Problemy Peredachi Informatsii, Vol. 17, no. 1, pp. 90-95, Jan.-March 1981.

[27] M. Georgiopoulos, L. Merakos, and P. Papantoni-Kazakos, "An asynchronous stack algorithm for CSMA and CSMA-CD channels," Proc. IEEE INFOCOM'85 Conference, Washington, D.C., March 1985.

[28] L. Merakos, and C. Bisdikian, "Delay analysis of the n-ary stack algorithm for a random-access broadcast channel," Proc. 22nd Allerton Conf. on Comm. Contr. Comp., Univ. of Illinois at Urbana-Champaign, Illinois, Oct. 1984. Also submitted to IEEE Trans. on Info. Theory.

[29] P. A. Humblet, "On the throughput of channel access algorithms with limited sensing," LIDS, Mass. Inst. of Technology, Dec. 1984, unpublished.

[30] L. Georgiadis and P. Papantoni-Kazakos, "A 0.487 throughput limited sensing algorithm," Technical Report EECS-TR-85-3, EECS Dept., Univ. of Connecticut, March 1985.

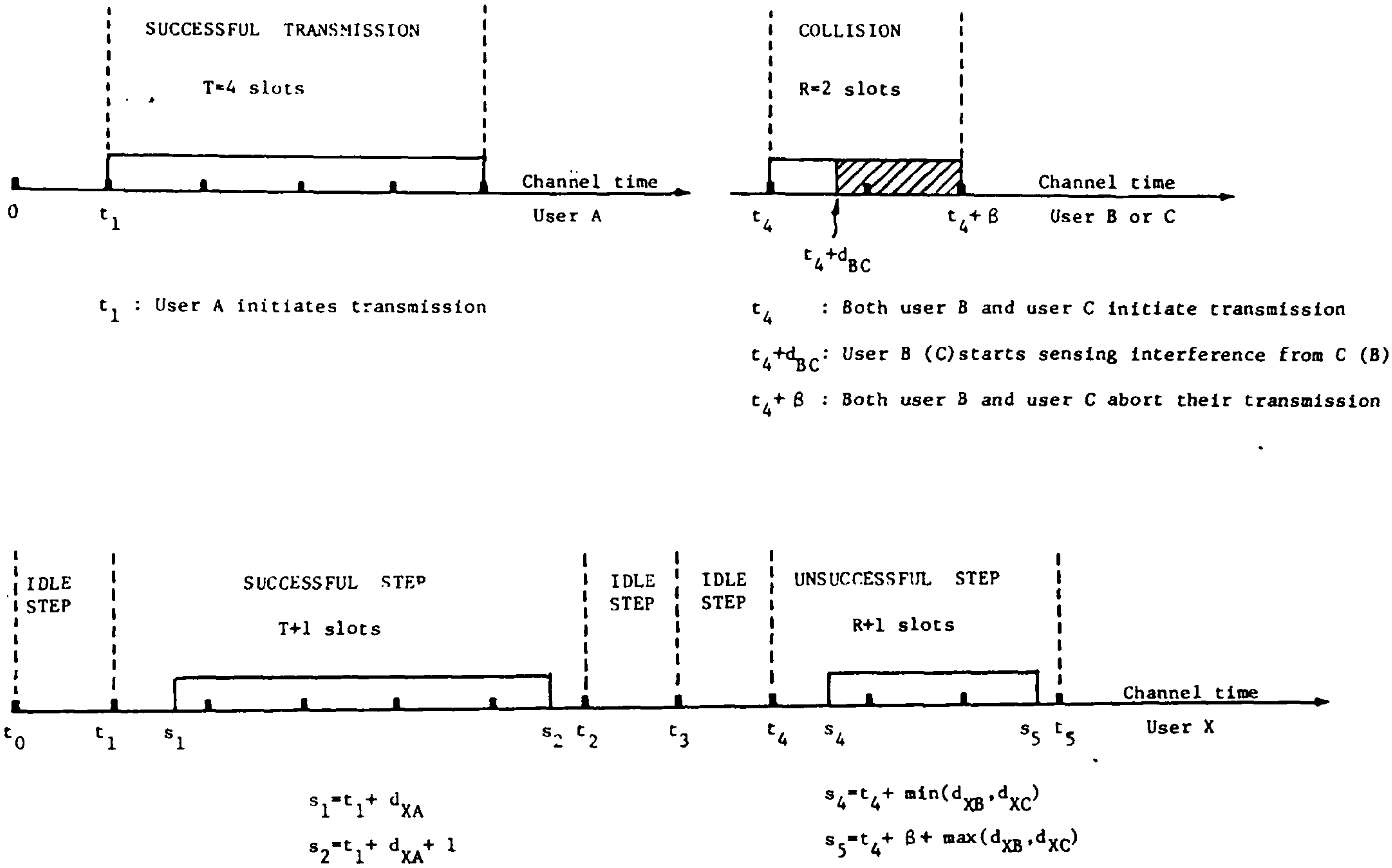


Figure 1. Channel activity induced by the LANSA, as perceived by an arbitrary user X.
Slot size = α (maximum propagation delay).
Packet transmission time = 1 (or $T = 1/\alpha$ slots)
Conflict truncation time = β (or $R = \beta/\alpha$ slots)
d_{XY} represents the propagation delay between user X and user Y.

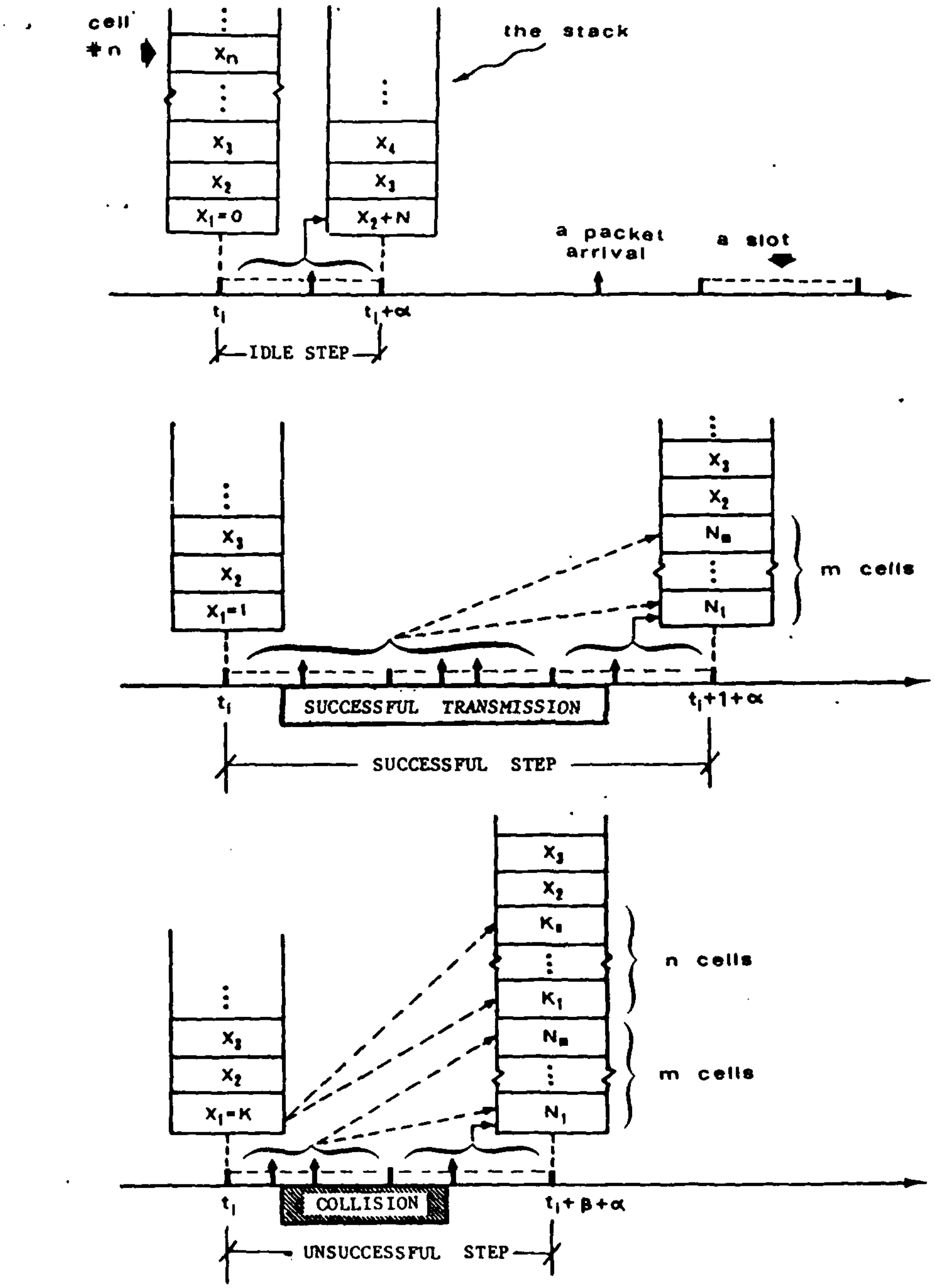


Figure 2. Illustration of the ith algorithm step using the stack. X_i denotes the number of packets in cell i, at t_i; $N_1+N_2+\ldots+N_m = N$, where N is the number of new arrivals, and $K_1+K_2+\ldots+K_n = K \geq 2$.

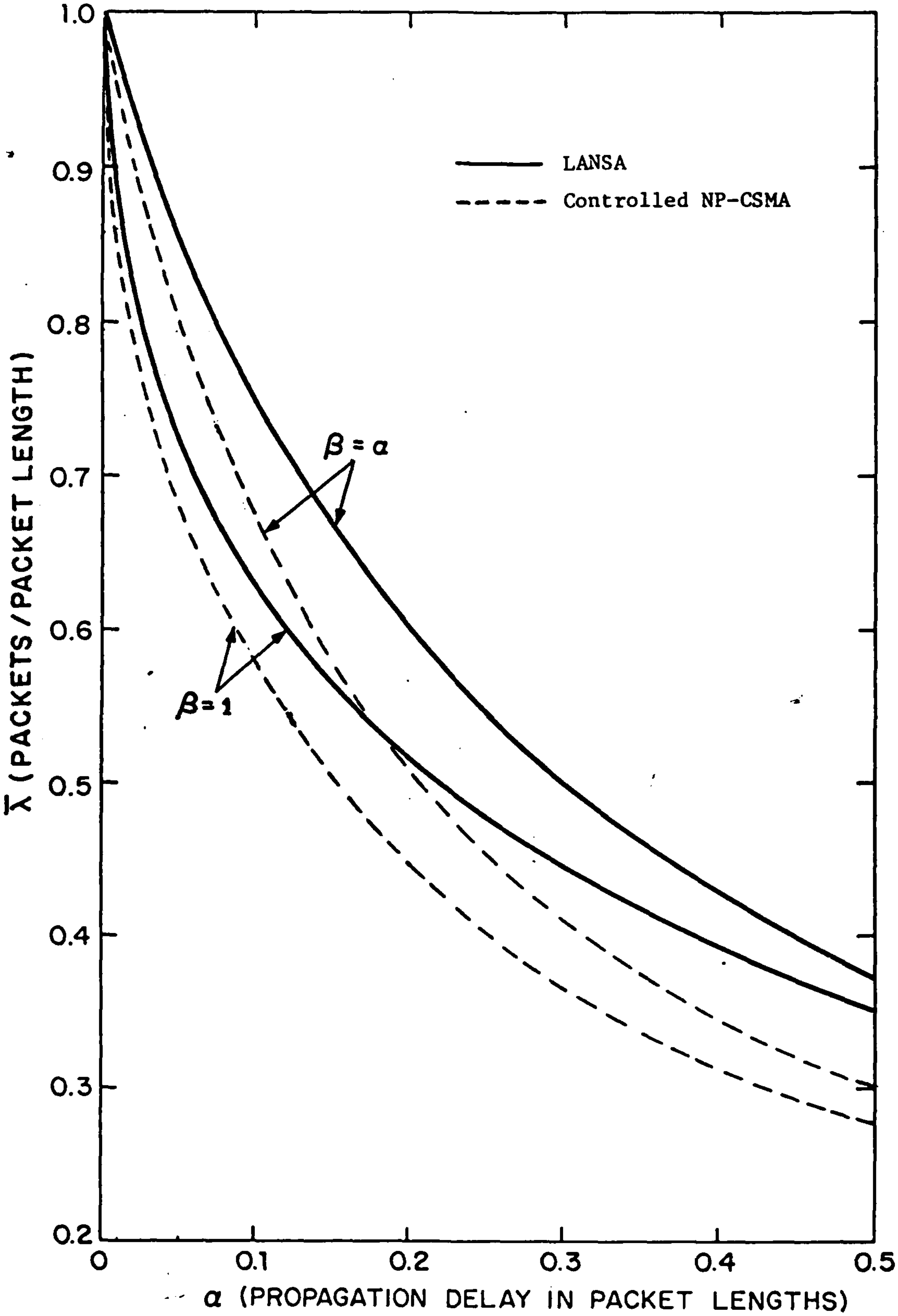


Figure 3. Lower bound $\bar{\lambda}$ on the maximum stable throughput for LANSA, and maximum stable throughput for controlled non-persistent CSMA (NP-CSMA).

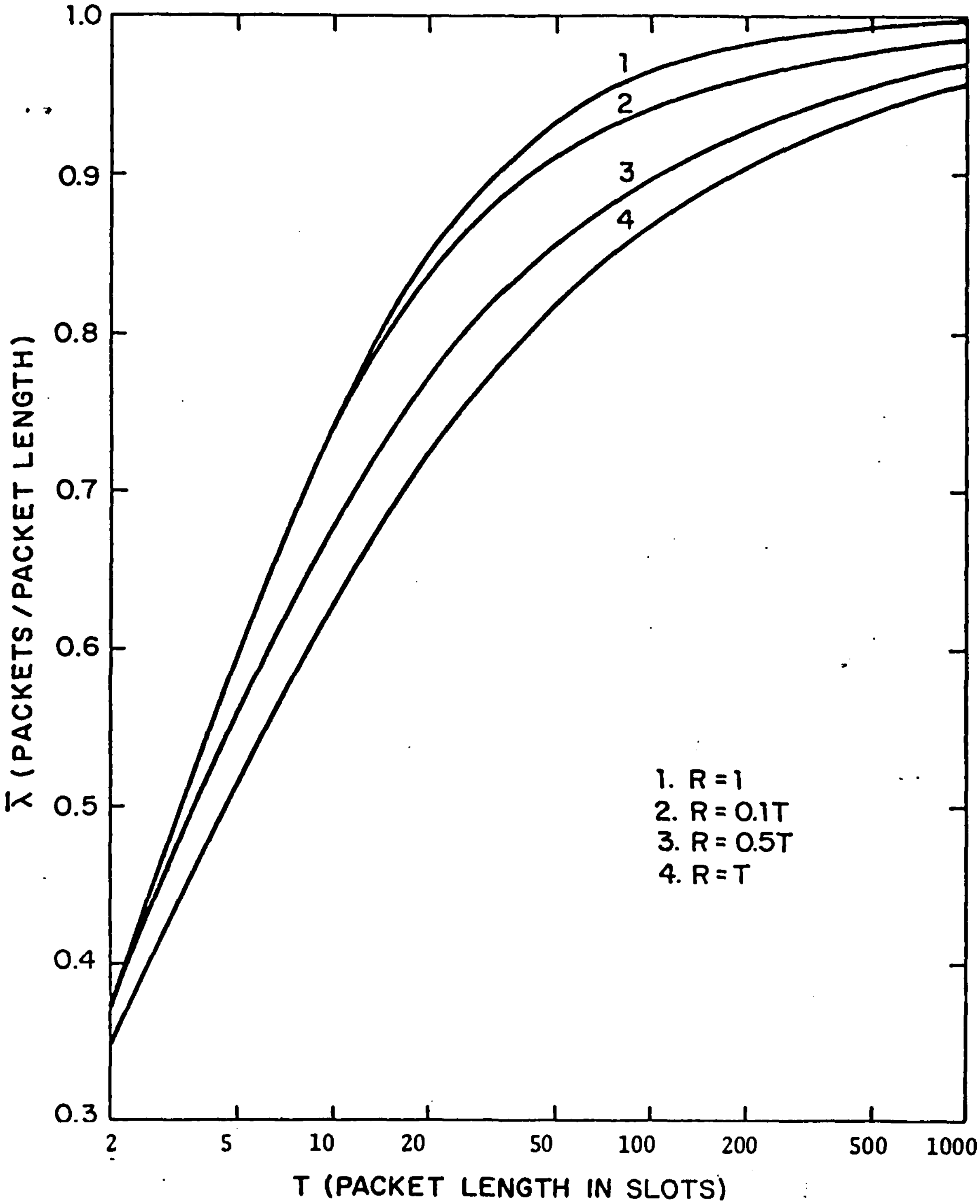


Figure 4 Lower bound $\bar{\lambda}$ on the maximum stable throughput for LANSA as a function of packet length with conflict truncation time as a parameter.

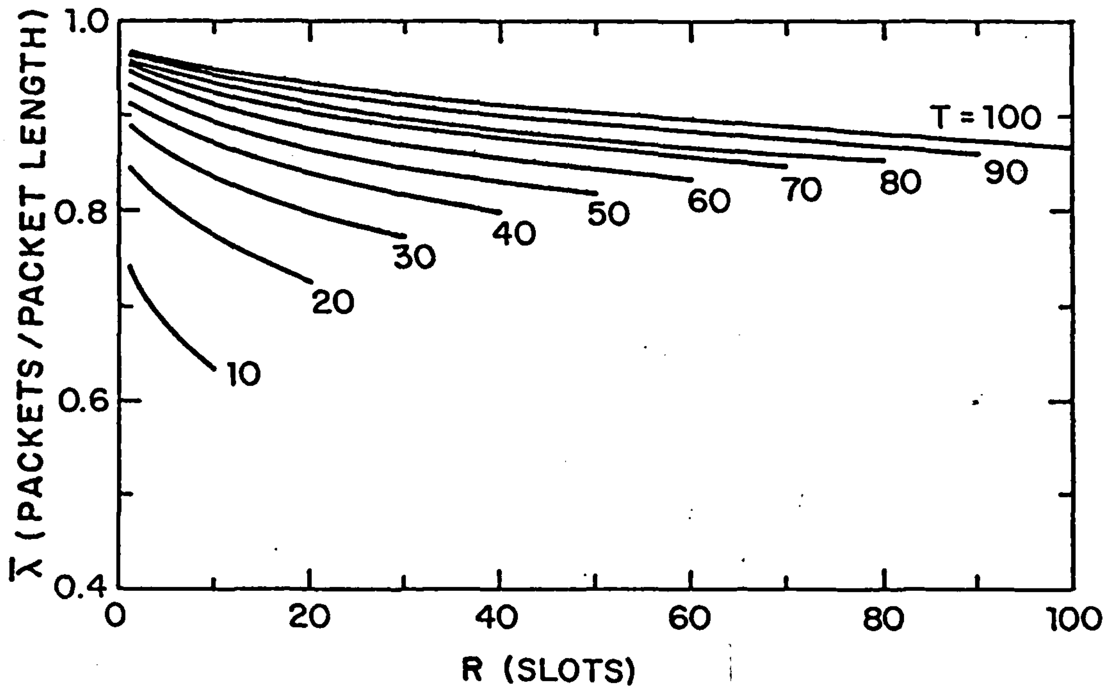


Figure 5 Lower bound $\bar{\lambda}$ on the maximum stable throughput for LANSA versus conflict truncation time with packet length as a parameter.

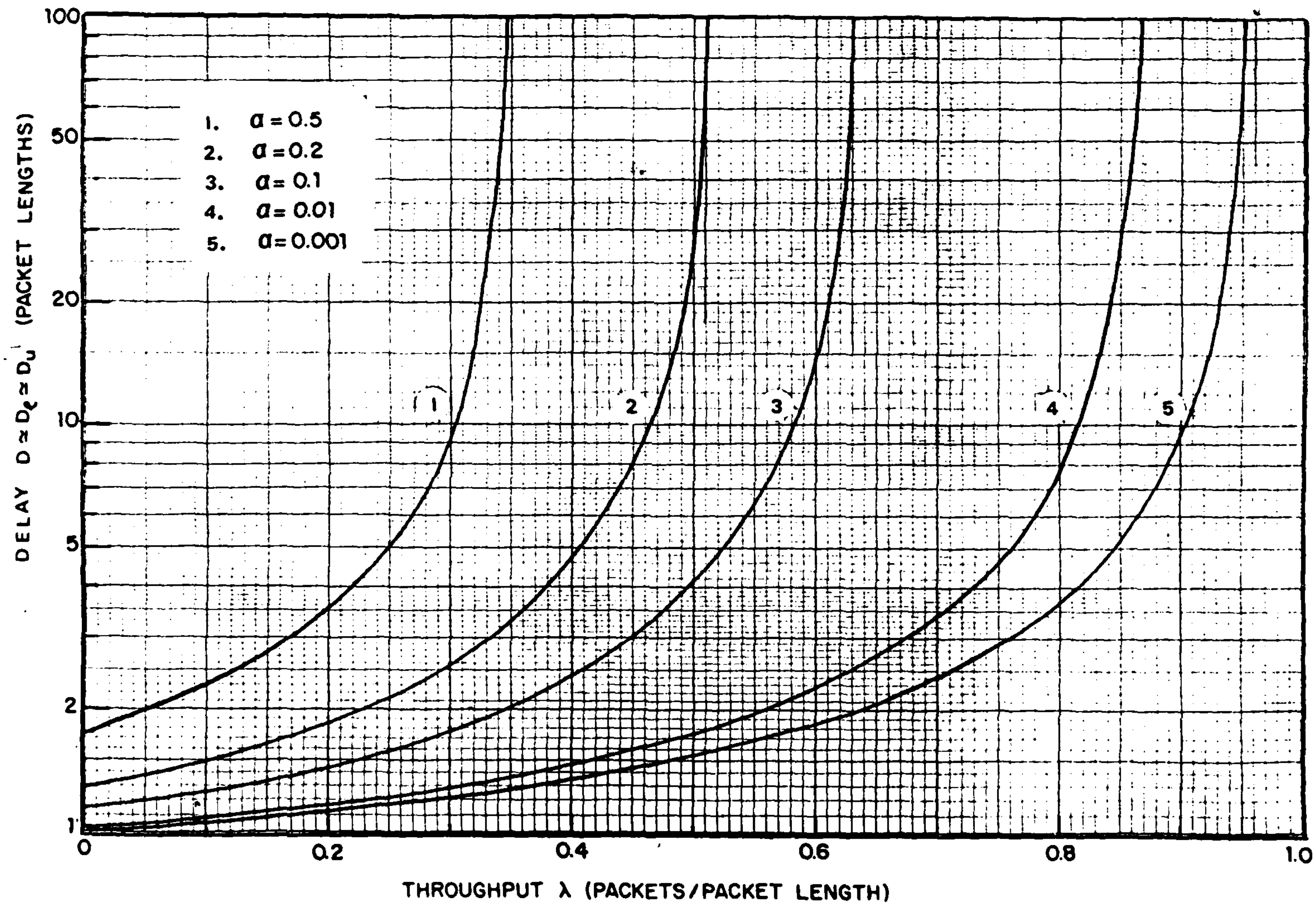


Figure 6. Mean packet delay versus throughput for LANSA ($\beta = 1$).

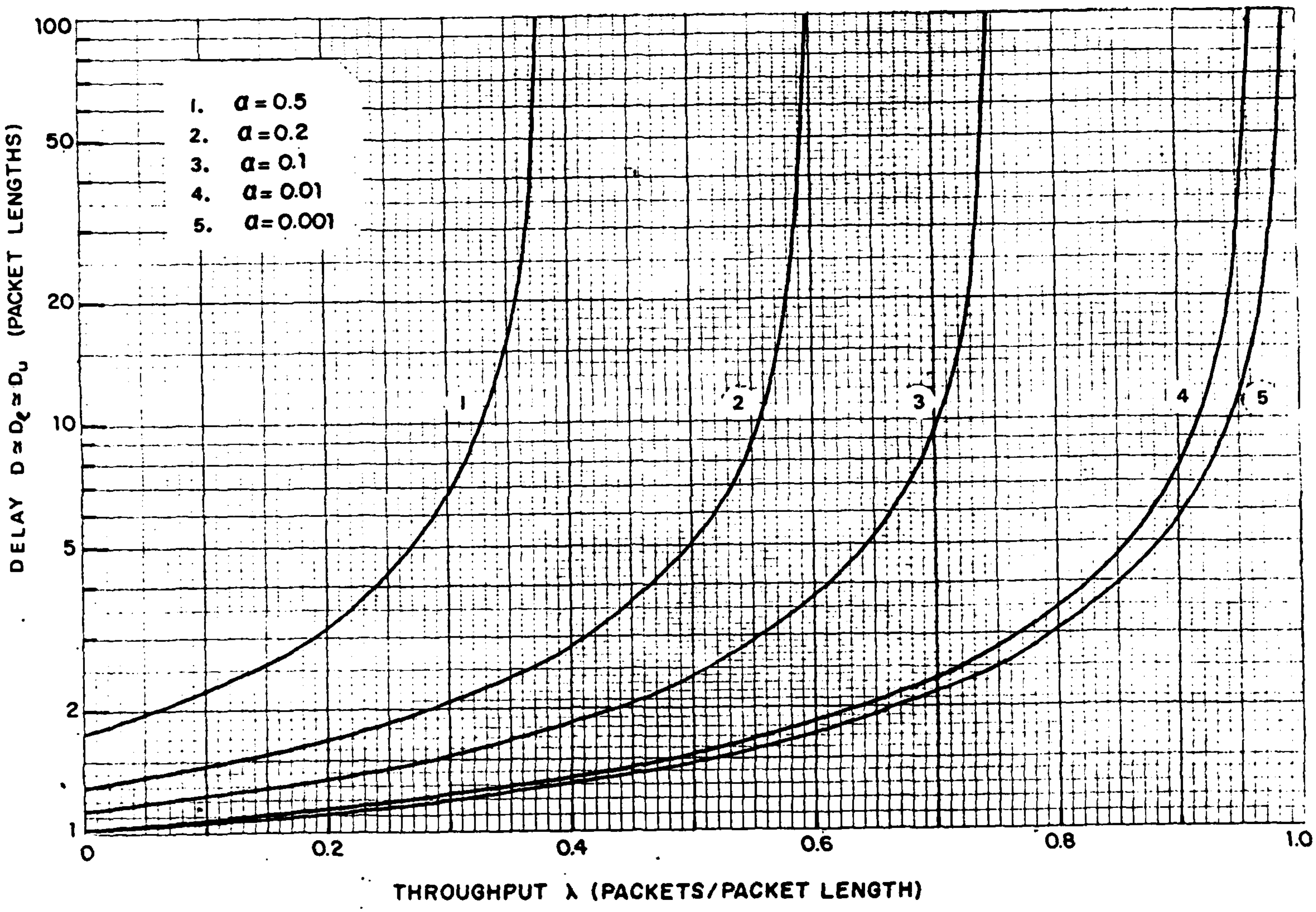


Figure 7. Mean packet delay versus throughput for LANSA (β = α).

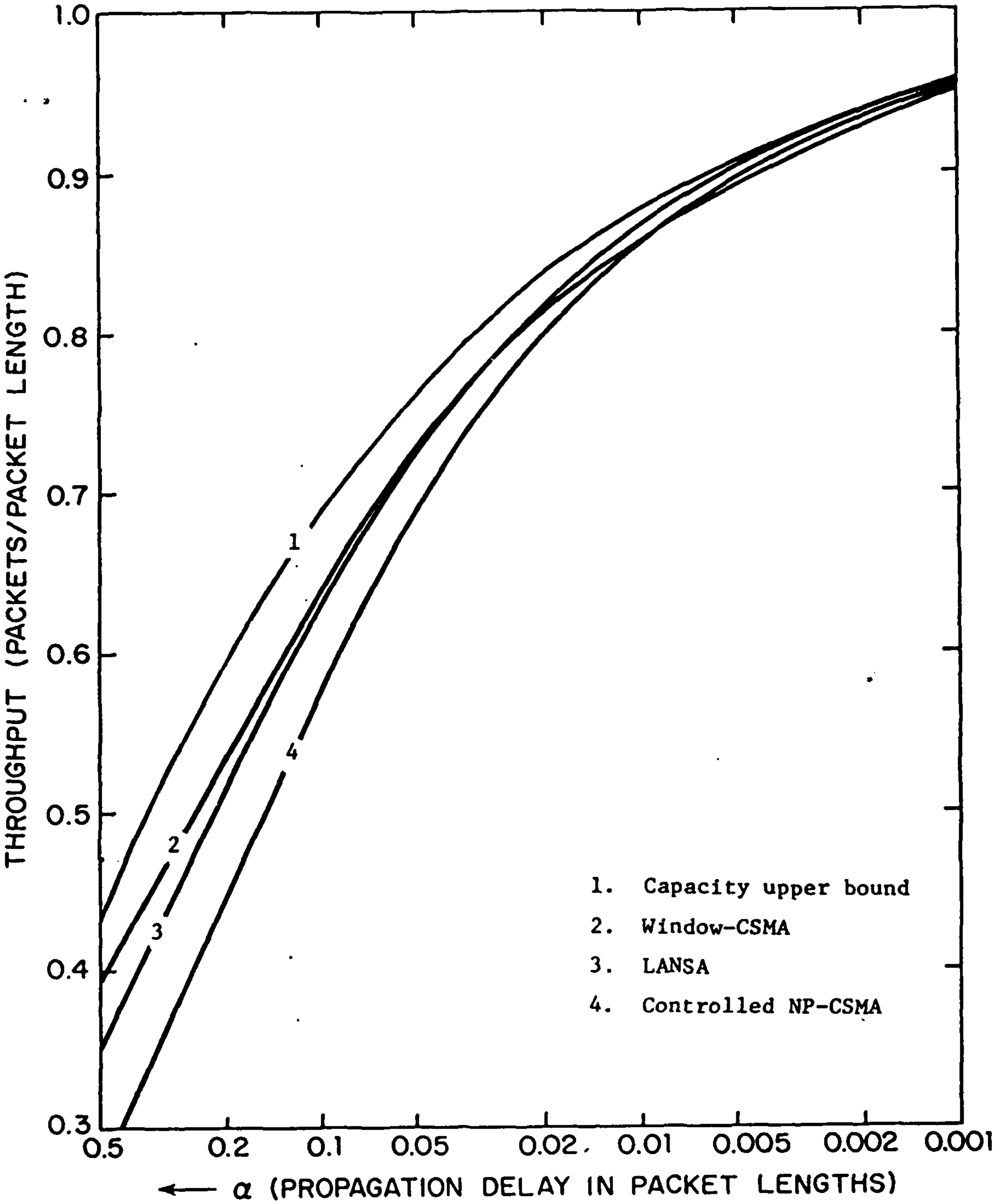


Figure 8. Lower bound $\bar{\lambda}$ on the maximum stable throughput for LANSA, maximum stable throughput for Window-CSMA and controlled NP-CSMA, and capacity upper bound, for $\beta = 1$.

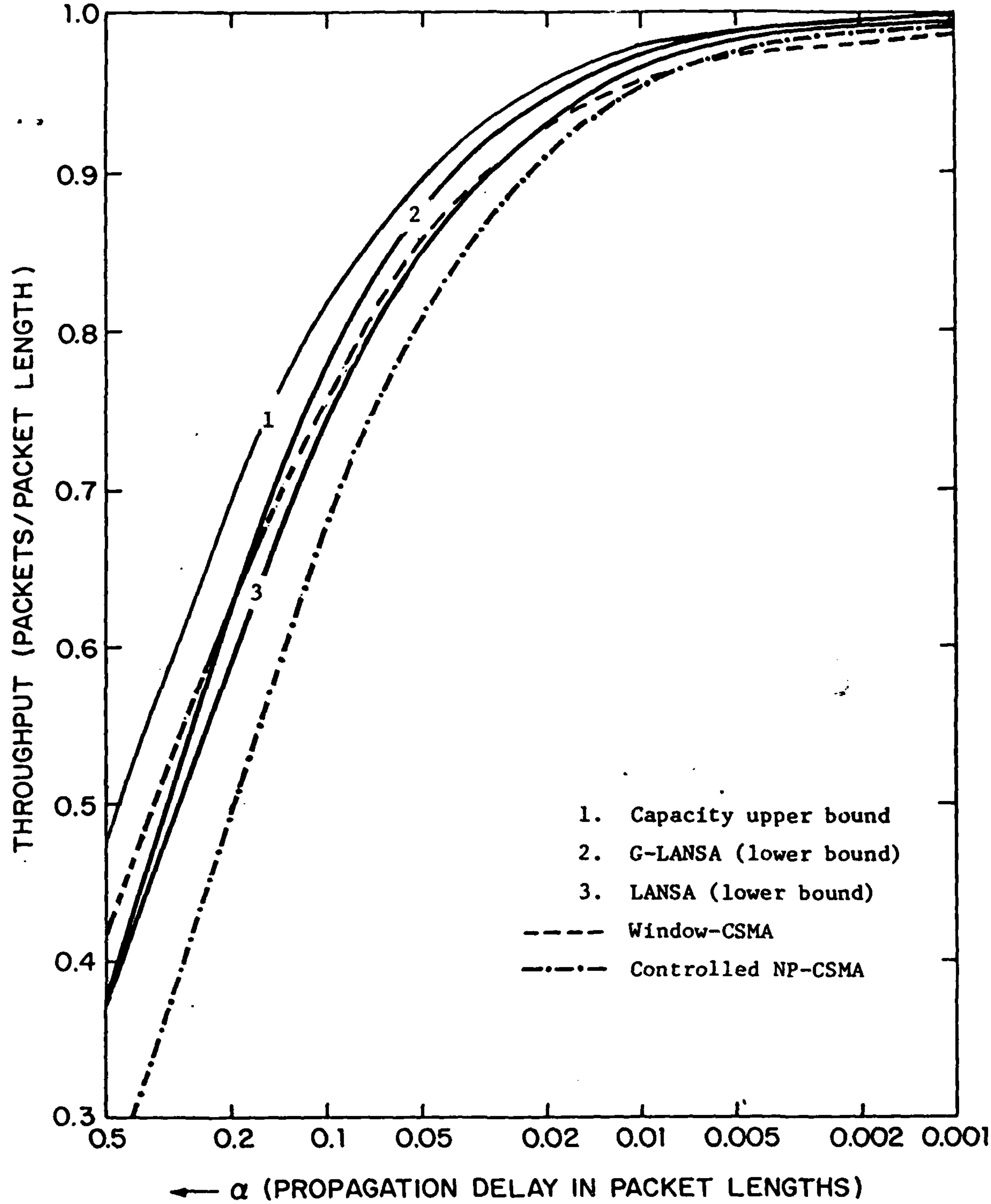


Figure 9. Lower bound $\tilde{\lambda}$ on the maximum stable throughput for LANSA and G-LANSA, maximum stable throughput for controlled NP-CSMA and Window-CSMA, and capacity upper bound versus propagation delay, for $\beta = \alpha$.

	LANSA $\beta = 1$			LANSA $\beta = 0.5$			LANSA $\beta = \alpha$		
α	$\bar{\lambda}$	m*	n*	$\bar{\lambda}$	m*	n*	$\bar{\lambda}$	m*	n*
0.500	0.348	1	4	0.371	1	3	0.371	1	3
0.400	0.390	1	4	0.419	1	4	0.427	1	3
0.300	0.442	1	4	0.480	1	4	0.498	1	3
0.200	0.510	2	4	0.561	1	4	0.597	1	3
0.100	0.629	2	5	0.676	2	4	0.743	1	3
0.050	0.723	3	6	0.771	2	5	0.849	1	3
0.020	0.816	5	9	0.855	4	7	0.931	1	3
0.010	0.867	7	12	0.898	5	9	0.964	1	3
0.005	0.904	10	16	0.928	7	12	0.981	1	3
0.002	0.938	16	24	0.954	11	17	0.992	1	3
0.001	0.956	22	33	0.968	16	24	0.996	1	3

Table 1. The lower bound $\bar{\lambda}$ on the maximum stable throughput of the LANSA, and the parameters m* , n* for representative values of the propagation delay α and the conflict truncation time β.

	λ	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.85	0.90	0.95
	k	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$	$L_k^u = L_k^\ell$
$\alpha=0.01$ $\beta=0.01$	1	1.1339	1.2778	1.4652	1.7190	2.0817	2.6421	3.6221	5.7725	8.2198	14.2901	54.9252
	2	2.3414	2.6401	3.0289	3.5554	4.3078	5.4704	7.5031	11.9636	17.0400	29.6309	113.9175
	3	3.5072	3.9552	4.5385	5.3282	6.4569	8.2010	11.2503	17.9412	25.5563	44.4438	170.8807
	4	4.6793	5.2776	6.0562	7.1108	8.6178	10.9460	15.0178	23.9515	34.1191	59.3373	228.1537
	5	5.8941	6.6029	7.5775	8.8973	10.7835	13.6980	18.7936	29.9749	42.7004	74.2631	285.5503
	6	7.0296	7.9289	9.0987	10.6850	12.9502	16.4512	22.5717	36.0019	51.2870	89.1981	342.9825
	7	8.2050	9.2550	10.6218	12.4725	15.1173	19.2043	26.3496	42.0286	59.8732	104.1322	400.4128
	8	9.3801	10.5807	12.1434	14.2595	17.2836	21.9566	30.1265	48.0538	68.4571	119.0631	457.8264
	9	10.5549	11.9060	13.6647	16.0461	19.4494	24.7082	33.9025	54.0774	77.0387	133.9894	515.2261
	10	11.7294	13.2310	15.1856	17.8323	21.6147	27.4593	37.6777	60.0998	85.6188	148.9135	572.6140
$\alpha=0.01$ $\beta=1.00$	1	1.2095	1.3781	1.6063	1.9323	2.4355	3.3124	5.2294	12.6831	45.4663	-	-
	2	3.8672	4.4122	5.1496	6.2025	7.8269	10.6599	16.8435	40.8946	146.6735	-	-
	3	5.2921	6.0390	7.0497	8.4927	10.7194	14.6025	23.0786	56.0467	201.0440	-	-
	4	6.8032	7.7642	9.0646	10.9212	13.7860	18.7822	29.6880	72.1066	258.6686	-	-
	5	8.3864	9.5716	11.1753	13.4650	16.9980	23.1598	36.6097	88.9241	319.0094	-	-
	6	10.0292	11.4469	13.3651	16.1041	20.3303	27.7010	43.7898	106.3682	381.5952	-	-
	7	11.7207	13.3777	15.6198	18.8212	23.7610	32.3761	51.1812	124.3253	446.0208	-	-
	8	13.4515	15.3534	17.9269	21.6013	27.2711	37.1593	58.7433	142.6961	511.9287	-	-
	9	15.2135	17.3647	20.2755	24.4315	30.8443	42.0284	66.4409	161.3954	579.0151	-	-
	10	16.9998	19.4036	22.6563	27.3004	34.4664	46.9640	74.2436	180.3502	647.0150	-	-

Table 2. Upper bound L_k^u and lower bound L_k^ℓ on L_k, for $\alpha = 0.01$ and $\beta = 0.01$, 1.00

LANSA	Upper bound $L_u^*(\lambda)$, and lower bound $L_\ell^*(\lambda)$ on the mean session length $L(\lambda)$							
	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$		$\alpha = 0.001$	
	$\beta = 0.5$	$\beta = 1$	$\beta = 0.1$	$\beta = 1$	$\beta = 0.01$	$\beta = 1$	$\beta = 0.001$	$\beta = 1$
λ ,	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$	$L_u^* \simeq L_\ell^*$
0.05	.542186	.542484	.105836	.106422	.001053	.010565	.001052	.001053
0.10	.597494	.599766	.112458	.113858	.001112	.011200	.001111	.001113
0.15	.675017	.682702	.120075	.122636	.001787	.011918	.001176	.001181
0.20	.793406	.815925	.128970	.133235	.012535	.012739	.001250	.001257
0.25	.998311	1.066750	.139539	.146374	.013387	.013688	.001333	.001343
0.30	1.439901	1.712785	.152345	.163195	.014366	.014797	.001429	.001443
0.35	3.077182	7.018270	.168229	.185622	.015502	.016115	.001539	.001559
0.40	-	-	.188501	.217179	.016837	.017708	.001668	.001696
0.45	-	-	.215316	.265082	.018429	.019673	.001820	.001859
0.50	-	-	.252507	.346803	.020360	.022164	.002003	.002058
0.55	-	-	.307594	.518380	.022753	.025427	.002227	.002306
0.60	-	-	.397642	1.111720	.025796	.029892	.002507	.002623
0.65	-	-	.571461	-	.029797	.036381	.002868	.003043
0.70	-	-	1.047590	-	.035291	.046691	.003351	.003626
0.75	-	-	-	-	.043310	.065628	.004030	.004492
0.80	-	-	-	-	.056113	.111880	.005053	.005909
0.85	-	-	-	-	.079806	.398382	.006775	.008658
0.90	-	-	-	-	.138564	-	.010279	.016273
0.95	-	-	-	-	.531879	-	.021297	.141640

Table 3.

LANSA	Upper bound $A_u^*(\lambda)$, and lower bound $A_\ell^*(\lambda)$ on the mean access delay $A(\lambda)$.							
	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$		$\alpha = 0.001$	
	$\beta=0.5$	$\beta=1$	$\beta=0.1$	$\beta=1$	$\beta=0.01$	$\beta=1$	$\beta=0.001$	$\beta=1$
λ	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$	$A_u^* \simeq A_\ell^*$
0.05	0.3800	0.3869	0.0964	0.0914	0.0321	0.0309	0.0257	0.0255
0.10	0.4100	0.4178	0.1228	0.1184	0.0572	0.0562	0.0507	0.0505
0.15	0.4400	0.4501	0.1492	0.1459	0.0824	0.0815	0.0757	0.0756
0.20	0.4700	0.4843	0.1756	0.1740	0.1075	0.1070	0.1007	0.1007
0.25	0.5000	0.5212	0.2020	0.2029	0.1327	0.1325	0.1257	0.1258
0.30	0.5300	0.5615	0.2284	0.2328	0.1578	0.1583	0.1507	0.1510
0.35	0.5600	-	0.2584	0.2637	0.1830	0.1843	0.1758	0.1763
0.40	-	-	0.2812	0.2960	0.2081	0.2104	0.2008	0.2015
0.45	--	-	0.3076	0.3298	0.2333	0.2369	0.2258	0.2268
0.50	-	-	0.3340	0.3653	0.2584	0.2636	0.2508	0.2523
0.55	-	-	0.3604	0.4029	0.2836	0.2907	0.2758	0.2778
0.60	-	-	0.3868	0.4428	0.3087	0.3181	0.3008	0.3034
0.65	-	-	0.4132	-	0.3339	0.3460	0.3258	0.3292
0.70	-	--	0.4396	-	0.3590	0.3742	0.3509	0.3550
0.75	-	-	-	-	0.3842	0.4030	0.3759	0.3810
0.80	-	-	-	-	0.4093	0.4322	0.4009	0.4071
0.85	-	-	-	-	0.4345	0.4620	0.4259	0.4334
0.90	-	-	-	-	0.4596	-	0.4509	0.4598
0.95	-	-	-	-	0.4848	-	0.4759	0.4864

Table 4.

LANSA	Upper bound $C_u^*(\lambda)$, and lower bound $C_\ell^*(\lambda)$ on the mean contention delay $C(\lambda)$.							
	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$		$\alpha = 0.001$	
	$\beta=0.5$	$\beta=1$	$\beta=0.1$	$\beta=1$	$\beta=0.01$	$\beta=1$	$\beta=0.001$	$\beta=1$
λ	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$	$C_u^* \simeq C_\ell^*$
0.05	1.6240	1.6461	1.1094	1.1207	1.0118	1.0144	1.0023	1.0031
0.10	1.8298	1.8917	1.1273	1.1554	1.0169	1.0235	1.0066	1.0086
0.15	2.1696	2.3101	1.1555	1.2081	1.0260	1.0383	1.0145	1.0181
0.20	2.7534	3.0687	1.1964	1.2852	1.0399	1.0600	1.0264	1.0324
0.25	3.8552	4.6375	1.2535	1.3963	1.0594	1.0904	1.0434	1.0526
0.30	6.3832	8.9758	1.3319	1.5568	1.0858	1.1317	1.0664	1.0799
0.35	16.1615	-	1.4392	1.7932	1.1208	1.1871	1.0968	1.1162
0.40	-	-	1.5870	2.1543	1.1667	1.2614	1.1365	1.1640
0.45	-	-	1.7951	2.7405	1.2264	1.3613	1.1881	1.2265
0.50	-	-	2.0981	3.7957	1.3046	1.4973	1.2552	1.3088
0.55	-	-	2.5644	6.1033	1.4078	1.6868	1.3429	1.4182
0.60	-	-	3.3489	14.3073	1.5462	1.9597	1.4591	1.5660
0.65	-	-	4.8955	-	1.7363	2.3735	1.6160	1.7713
0.70	-	-	9.1909	-	2.0071	3.0535	1.8343	2.0676
0.75	-	-	-	-	2.4143	4.3357	2.1512	2.5213
0.80	-	-	-	-	3.0799	7.5258	2.6421	3.2834
0.85	-	-	-	-	4.3337	27.5038	3.4848	4.7887
0.90	-	-	-	-	7.4811	-	5.2257	9.0118
0.95	-	-	-	-	28.6961	-	10.7536	79.0137

Table 5.

	G-LANSA $\beta = 1$					G-LANSA $\beta = 0.5$					G-LANSA $\beta = \alpha$				
α	$\bar{\lambda}$	m^*	$\bar{m}^*$	n^*	p^*	$\bar{\lambda}$	m^*	$\bar{m}^*$	n^*	p^*	$\bar{\lambda}$	m^*	$\bar{m}^*$	n^*	p^*
0.5	0.348	1	1	4	0	0.375	1	1	3	0.1	0.375	1	1	3	0.1
0.4	0.390	1	1	4	0	0.423	1	1	3	0.1	0.431	1	1	3	0.1
0.3	0.443	1	1	4	0	0.483	1	1	3	0.1	0.504	1	1	3	0.2
0.2	0.510	2	2	4	0	0.561	1	1	3	0.1	0.608	1	1	2	0.3
0.1	0.629	2	2	5	0	0.682	2	1	4	0	0.758	1	1	2	0.3
0.05	0.723	3	3	6	0	0.772	2	1	5	0	0.862	1	3	0	1
0.02	0.816	5	5	9	0	0.856	4	2	6	0	0.940	1	3	0	1
0.01	0.867	7	7	12	0	0.899	5	3	8	0	0.969	1	3	0	1
0.005	0.904	10	10	16	0	0.928	7	3	11	0	0.984	1	3	0	1
0.002	0.938	16	16	24	0	0.955	11	5	14	0	0.993	1	3	0	1
0.001	0.956	22	22	33	0	0.968	15	7	19	0	0.996	1	3	0	1

Table 6. The lower bound $\bar{\lambda}$ on the maximum stable throughput of the G-LANSA (rule 3') , and the parameters m^* , n^* , $\bar{m}^*$, p^* for representative values of the propagation delay α and the conflict truncation time β.

	G-LANSA (rule 3") $\beta = \alpha$				
α	$\bar{\lambda}$	m^*	$\bar{m}^*$	n^*	$p^*=\bar{p}^*$
0.500	0.378	1	1	1	0.32
0.200	0.625	1	1	1	0.34
0.100	0.777	1	1	1	0.35
0.050	0.877	1	1	1	0.36
0.020	0.947	1	1	1	0.36
0.010	0.973	1	1	1	0.36
0.005	0.985	1	1	1	0.36
0.002	0.993	1	1	1	0.37
0.001	0.996	1	1	1	0.37

Table 7. The lower bound $\bar{\lambda}$ on the maximum stable throughput of the G-LANSA (rule 3"), and the parameters m^* , $\bar{m}^*$, n^* , p^* , $\bar{p}^*$ for representative values of the propagation delay α and for $\beta = \alpha$.

END
DTIC
8-86